

# Lab 6: Ciphers and Digital Certificates

---

In this lab we will investigate a range of ciphers and also how we can view the details within digital certificates.

## A Ciphers

---

Use your desktop computer to complete the following:

| No | Description   | Result  |
|----|---|---|
| 1  | Go to:<br><br><b><a href="http://asecuritysite.com/Challenges">http://asecuritysite.com/Challenges</a></b><br><br>Click on the “Start Challenge” button, and see if you can score over 30 points. | Your score:   |
| 2  | Using: <a href="http://asecuritysite.com/Encryption/testprime">http://asecuritysite.com/Encryption/testprime</a> Test for the following prime numbers:  | 91: [Yes] [No]<br>421: [Yes] [No]<br>1449: [Yes] [No] |
| 3  | Using: <a href="http://asecuritysite.com/Encryption/gcd">http://asecuritysite.com/Encryption/gcd</a> Determine the GCD for the following:   | 88, 46:<br><br>105, 35:                               |
| 4  | Using: <a href="http://asecuritysite.com/coding/ascii">http://asecuritysite.com/coding/ascii</a><br><br>Determine the Base 64 and Hex values for the following strings:                           | Hello: hello:<br><br>HELLO:                           |

|          |   |  |
|----------|---|--|
| <b>5</b> | <p>Using: <a href="http://asecuritysite.com/coding/ascii">http://asecuritysite.com/coding/ascii</a></p> <p>Determine the following ASCII strings for these encoded formats:</p> | <p>bGxveWRz</p> <p>6E6170696572</p> <p>01000001 01101110 01101011<br/>01101100 01100101 00110001<br/>00110010 00110011</p> |
| <b>6</b> | <p>Using: <a href="http://asecuritysite.com/Coding/exor">http://asecuritysite.com/Coding/exor</a></p> <p>Determine the EX-OR of “hello” ex-ORed with the letter ‘t’</p>         | <p>Hex: Base 64:</p> <p>Is the result printable in ASCII?<br/>[Yes][No]</p>  |
| <b>7</b> | <p>What is the result of <math>53,431 \bmod 453</math>?</p>   |  |
| <b>8</b> | <p>Generate a random hex number from: <a href="http://asecuritysite.com/Encryption/js01">http://asecuritysite.com/Encryption/js01</a></p>                                       | <p>How many hex<br/>characters does the<br/>result have?</p>   |

|   |  |   |
|---|--|---|
| 9 | <p>Try and crack some certificates from:</p> <p><a href="http://asecuritysite.com/Encryption/certcrack">http://asecuritysite.com/Encryption/certcrack</a></p> <p>What are the passwords for 'bill09.pfx', 'bill18.pfx', and 'country04.pfx'?</p> | <p>bill09.pfx:</p> <p>bill18.pfx:</p> <p>country04.pfx:</p> |
|---|--|---|

10. We can also create a short **Python script** to try to crack the same certificates.

Boot up your Kali VM on your public network, and download the following archive:

<http://asecuritysite.com/public/certs.zip>

Extract the certificates into the /root folder, and then move into that folder. Now use openssl to try a password:

```
openssl pkcs12 -nokeys -in bill01.pfx -passin pass:orange
```

Did you manage to run the script?

What password is correct for bill01.pfx?

Now implement the Python script given below:

```
from OpenSSL import crypto
words=[]
words.append("coconut")
words.append("mango")
words.append("apples")
words.append("apple")
words.append("oranges")
words.append("orange")
words.append("ankle")
words.append("password")
words.append("bill")
words.append("battery")
```

```
for passwd in words:
    try:
        p12 = crypto.load_pkcs12(open("fredpfx.pfx", 'rb').read(), passwd)
        certificate = p12.get_certificate()
        p12.get_privatekey()

        print certificate.get_serial_number()
        print certificate.get_issuer().get_components()
        print certificate.get_signature_algorithm()
        print ("Success: "+passwd)
    except Exception as ex:
        print (".")
```

Can adapt this script to crack some of the other certificates contained in the archive you have downloaded. Bill01.pdf to bill18.pdf are based on fruits (in lowercase), country01.pdf to country06.pdf are based on countries.

Outline the passwords of the certificates:

Can you modify the code so that it shows other details from the certificate, such as its public key, subject, version and “notBefore”, and “notAfter”.

Ref: <https://pyopenssl.org/en/0.15.1/api/crypto.html#x509name-objects>

## B Frequency Analysis

---

Now see if you can crack the **five minute cracking challenge** for: <http://asecuritysite.com/challenges/scramb>

## C Character Mapping

---

Complete the following table for each of the characters:

| Char    | Decimal | Binary | Hex | Oct | HTML |
|---------|---------|--------|-----|-----|------|
| (Space) |         |        |     |     |      |
| a       |         |        |     |     |      |
| }       |         |        |     |     |      |
| Ã       |         |        |     |     |      |
| ÿ       |         |        |     |     |      |

## D Test

---

1. Crack some Caesar codes at: <http://asecuritysite.com/tests/tests?sortBy=caesar>
2. Determine some hex conversions at: <http://asecuritysite.com/tests/tests?sortBy=hex01>
3. Determine some Base64 conversions: <http://asecuritysite.com/tests/tests?sortBy=ascii01>

## Shifted alphabet

---

|    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1  | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 2  | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| 3  | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| 4  | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| 5  | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| 6  | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| 7  | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| 8  | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| 9  | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| 10 | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| 11 | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| 12 | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| 13 | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| 14 | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| 15 | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| 16 | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| 17 | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| 18 | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| 19 | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| 20 | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| 21 | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| 22 | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| 23 | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| 24 | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| 25 | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| 26 | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

## ASCII table

| Char  | Dec | Oct  | Hex  | Char | Dec | Oct  | Hex  | Char | Dec | Oct  | Hex  | Char  | Dec | Oct  | Hex  |
|-------|-----|------|------|------|-----|------|------|------|-----|------|------|-------|-----|------|------|
| (nul) | 0   | 0000 | 0x00 | (sp) | 32  | 0040 | 0x20 | @    | 64  | 0100 | 0x40 | `     | 96  | 0140 | 0x60 |
| (soh) | 1   | 0001 | 0x01 | !    | 33  | 0041 | 0x21 | A    | 65  | 0101 | 0x41 | a     | 97  | 0141 | 0x61 |
| (stx) | 2   | 0002 | 0x02 | "    | 34  | 0042 | 0x22 | B    | 66  | 0102 | 0x42 | b     | 98  | 0142 | 0x62 |
| (etx) | 3   | 0003 | 0x03 | #    | 35  | 0043 | 0x23 | C    | 67  | 0103 | 0x43 | c     | 99  | 0143 | 0x63 |
| (eot) | 4   | 0004 | 0x04 | \$   | 36  | 0044 | 0x24 | D    | 68  | 0104 | 0x44 | d     | 100 | 0144 | 0x64 |
| (enq) | 5   | 0005 | 0x05 | %    | 37  | 0045 | 0x25 | E    | 69  | 0105 | 0x45 | e     | 101 | 0145 | 0x65 |
| (ack) | 6   | 0006 | 0x06 | &    | 38  | 0046 | 0x26 | F    | 70  | 0106 | 0x46 | f     | 102 | 0146 | 0x66 |
| (bel) | 7   | 0007 | 0x07 | '    | 39  | 0047 | 0x27 | G    | 71  | 0107 | 0x47 | g     | 103 | 0147 | 0x67 |
| (bs)  | 8   | 0010 | 0x08 | (    | 40  | 0050 | 0x28 | H    | 72  | 0110 | 0x48 | h     | 104 | 0150 | 0x68 |
| (ht)  | 9   | 0011 | 0x09 | )    | 41  | 0051 | 0x29 | I    | 73  | 0111 | 0x49 | i     | 105 | 0151 | 0x69 |
| (nl)  | 10  | 0012 | 0x0a | *    | 42  | 0052 | 0x2a | J    | 74  | 0112 | 0x4a | j     | 106 | 0152 | 0x6a |
| (vt)  | 11  | 0013 | 0x0b | +    | 43  | 0053 | 0x2b | K    | 75  | 0113 | 0x4b | k     | 107 | 0153 | 0x6b |
| (np)  | 12  | 0014 | 0x0c | ,    | 44  | 0054 | 0x2c | L    | 76  | 0114 | 0x4c | l     | 108 | 0154 | 0x6c |
| (cr)  | 13  | 0015 | 0x0d | -    | 45  | 0055 | 0x2d | M    | 77  | 0115 | 0x4d | m     | 109 | 0155 | 0x6d |
| (so)  | 14  | 0016 | 0x0e | .    | 46  | 0056 | 0x2e | N    | 78  | 0116 | 0x4e | n     | 110 | 0156 | 0x6e |
| (si)  | 15  | 0017 | 0x0f | /    | 47  | 0057 | 0x2f | O    | 79  | 0117 | 0x4f | o     | 111 | 0157 | 0x6f |
| (dle) | 16  | 0020 | 0x10 | 0    | 48  | 0060 | 0x30 | P    | 80  | 0120 | 0x50 | p     | 112 | 0160 | 0x70 |
| (dc1) | 17  | 0021 | 0x11 | 1    | 49  | 0061 | 0x31 | Q    | 81  | 0121 | 0x51 | q     | 113 | 0161 | 0x71 |
| (dc2) | 18  | 0022 | 0x12 | 2    | 50  | 0062 | 0x32 | R    | 82  | 0122 | 0x52 | r     | 114 | 0162 | 0x72 |
| (dc3) | 19  | 0023 | 0x13 | 3    | 51  | 0063 | 0x33 | S    | 83  | 0123 | 0x53 | s     | 115 | 0163 | 0x73 |
| (dc4) | 20  | 0024 | 0x14 | 4    | 52  | 0064 | 0x34 | T    | 84  | 0124 | 0x54 | t     | 116 | 0164 | 0x74 |
| (nak) | 21  | 0025 | 0x15 | 5    | 53  | 0065 | 0x35 | U    | 85  | 0125 | 0x55 | u     | 117 | 0165 | 0x75 |
| (syn) | 22  | 0026 | 0x16 | 6    | 54  | 0066 | 0x36 | V    | 86  | 0126 | 0x56 | v     | 118 | 0166 | 0x76 |
| (etb) | 23  | 0027 | 0x17 | 7    | 55  | 0067 | 0x37 | W    | 87  | 0127 | 0x57 | w     | 119 | 0167 | 0x77 |
| (can) | 24  | 0030 | 0x18 | 8    | 56  | 0070 | 0x38 | X    | 88  | 0130 | 0x58 | x     | 120 | 0170 | 0x78 |
| (em)  | 25  | 0031 | 0x19 | 9    | 57  | 0071 | 0x39 | Y    | 89  | 0131 | 0x59 | y     | 121 | 0171 | 0x79 |
| (sub) | 26  | 0032 | 0x1a | :    | 58  | 0072 | 0x3a | Z    | 90  | 0132 | 0x5a | z     | 122 | 0172 | 0x7a |
| (esc) | 27  | 0033 | 0x1b | ;    | 59  | 0073 | 0x3b | [    | 91  | 0133 | 0x5b | {     | 123 | 0173 | 0x7b |
| (fs)  | 28  | 0034 | 0x1c | <    | 60  | 0074 | 0x3c | \    | 92  | 0134 | 0x5c |       | 124 | 0174 | 0x7c |
| (gs)  | 29  | 0035 | 0x1d | =    | 61  | 0075 | 0x3d | ]    | 93  | 0135 | 0x5d | }     | 125 | 0175 | 0x7d |
| (rs)  | 30  | 0036 | 0x1e | >    | 62  | 0076 | 0x3e | ^    | 94  | 0136 | 0x5e | ~     | 126 | 0176 | 0x7e |
| (us)  | 31  | 0037 | 0x1f | ?    | 63  | 0077 | 0x3f | _    | 95  | 0137 | 0x5f | (del) | 127 | 0177 | 0x7f |

## Base 64

---

Example:

“fred”            01100110   01110010   01100101   01100100  
Split into 6 bits: 011001   100111   001001   100101   011001   00  
                     Z            n            J            l            Z            A    =    =

| Val | Encoding | Val | Encoding | Val | Encoding | Val | Encoding |
|-----|----------|-----|----------|-----|----------|-----|----------|
| 0   | A        | 16  | Q        | 32  | g        | 48  | w        |
| 1   | B        | 17  | R        | 33  | h        | 49  | x        |
| 2   | C        | 18  | S        | 34  | i        | 50  | y        |
| 3   | D        | 19  | T        | 35  | j        | 51  | z        |
| 4   | E        | 20  | U        | 36  | k        | 52  | 0        |
| 5   | F        | 21  | V        | 37  | l        | 53  | 1        |
| 6   | G        | 22  | W        | 38  | m        | 54  | 2        |
| 7   | H        | 23  | X        | 39  | n        | 55  | 3        |
| 8   | I        | 24  | Y        | 40  | o        | 56  | 4        |
| 9   | J        | 25  | Z        | 41  | p        | 57  | 5        |
| 10  | K        | 26  | a        | 42  | q        | 58  | 6        |
| 11  | L        | 27  | b        | 43  | r        | 59  | 7        |
| 12  | M        | 28  | c        | 44  | s        | 60  | 8        |
| 13  | N        | 29  | d        | 45  | t        | 61  | 9        |
| 14  | O        | 30  | e        | 46  | u        | 62  | +        |
| 15  | P        | 31  | f        | 47  | v        | 63  | /        |