Crypto basics 1

Choose the “key”
1. Plugboard settings: For example: A/L – P/R – T/D – B/W – K/F – O/Y

Each plug wire can connect two letters to be a pair. The two letters in a pair will swap over.

2. Rotor (or scrambler) arrangement: For example: 2 — 3 —1
3. Rotor orientations: For example: D – K –P

Each different combination of rotors would produce a different encoding scheme.
Each takes in a letter and outputs it as a different one.
When the first rotor has turned through all 26 positions, the second rotor clicks round, and when that’s made it round all the way, the third does the same

4. Choose reflector(During the Second World War two types of reflector, B and C, were used)

After a letter passes through all three rotors, it bounces off a “reflector” at the end, and passes back through all three rotors in the other direction.

Crypto basics 2

5*4*3=60 - ways to configure the five rotors.

26*26*26=17,576 - choices for initial configurations of the rotors’ numbers/letters

$$\frac{26!}{6! * 10! * 2^{10}} = 150,738,274,937,250$$

26 letters in the alphabet - 26! ways to arrange the letters
6 letters do not participate (because there are only 10 wires connected to the plugboard)
It does not matter what order the pairs are in
Order of the letters in the pair does not matter

Also there are 3 reflectors

Wrapping up:
60*17576*150738274937250.0*3=4.7688766565347905e+20

Nearly the same thing with Viola, here's the python script:

```
from math import factorial
def get_num_of_variants(num_characters, num_of_wires,
num_of_reflectors,
num_reflector_slots,
num_rotors,
num_rotors_slots):
  a = factorial(num_rotors
  )/factorial(num_rotors-num_rotors_slots)
  b= num_characters**num_rotors_slots
  c = factorial(num_characters)/(
  factorial(num_characters-num_of_wires*2)
  *factorial(num_of_wires)
  *2**num_of_wires)
  d=factorial(num_of_reflectors)/(
  factorial(num_of_reflectors-num_reflector_slots))
  return (a*b*c*d)
#Viola
num_rotors=50
num_of_reflectors=5
num_reflector_slots=1
num_rotors_slots=10
num_characters=30
summ=0
for num_of_wires in range(16):
  result = get_num_of_variants(num_characters, num_of_wires,
num_of_reflectors,
  num_reflector_slots, num_rotors, num_rotors_slots)
  summ = summ + result
print(summ)
#Enigma
num_rotors=5
num_of_reflectors=3
num_reflector_slots=1
num_rotors_slots=3
num_characters=26
num_of_wires=10
result = get_num_of_variants(num_characters, num_of_wires, num_of_reflectors,
num_reflector_slots, num_rotors, num_rotors_slots)
print(result)
```

The output of this script is

Viola:

6.679467659284613e+49

Enigma:
4.7688766565347905e+20

Crypto basics 3

1) Most popular ones are John the Ripper and hashcat
2) I think they are both great tools. The main difference is that hashcat uses GPU cracking(oclhashcat) while John uses CPU. I use hashcat, mostly because It is more familiar to me.

Crypto basics 4

6q7xpw/2.$la4KiUz87ohdszbOVoIopy2VTwm/5jEXvWSdWynh0CnP5T.MnJfVNCzp3IfJM
HUNuBhr1ewcYd8PyeKHqHQoe.:17770:0:99999:7:::
It is definitely from the /etc/shadow because of the format:

1. **Username** : It is your login name.
2. **Password** : It is your encrypted password. The password should be minimum 8-12 characters long including special characters, digits, lower case alphabetic and more. Usually password format is set to idsalt$hashed, The $id is the algorithm used On GNU/Linux as follows:
    1. **1** is MD5
    2. **$2a$** is Blowfish
    3. **$2y$** is Blowfish
    4. **5** is SHA-256
    5. **6** is SHA-512
3. **Last password change (lastchanged)** : Days since Jan 1, 1970 that password was last changed
4. **Minimum** : The minimum number of days required between password changes i.e. the number of days left before the user is allowed to change his/her password
5. **Maximum** : The maximum number of days the password is valid (after that user is forced to change his/her password)
6. **Warn** : The number of days before password is to expire that user is warned that his/her password must be changed
7. **Inactive** : The number of days after password expires that account is disabled

8. **Expire** : days since Jan 1, 1970 that account is disabled i.e. an absolute date specifying when the login may no longer be used.

From the point 2, we know that **6** is for SHA-512

Next step is to use hashcat:

```
hashcat -m 1800 -a 0 -o cracked.txt forhashmap.txt rockyou.txt --force
```

Crypto basics 5

	Block Cipher	Stream Cipher
1.	Block Cipher Converts the plain text into cipher text by taking plain text's block at a time.	Stream Cipher Converts the plaint text into cipher text by taking 1 byte of plain text at a time.
2.	Block cipher uses either 64 bits or more than 64 bits.	While stream cipher uses 8 bits.
3.	The complexity of block cipher is simple.	While stream cipher is more complex.
4.	Block cipher Uses confusion as well as diffusion.	While stream cipher uses only confusion.
5.	In block cipher, reverse encrypted text is hard.	While in stream cipher, reverse encrypted text is easy.
6.	The algorithm modes which are used in block cipher are: ECB (Electronic Code Book) and CBC (Cipher Block Chaining).	The algorithm modes which are used in stream cipher are: CFB (Cipher Feedback) and OFB (Output Feedback).

Crypto basics 6

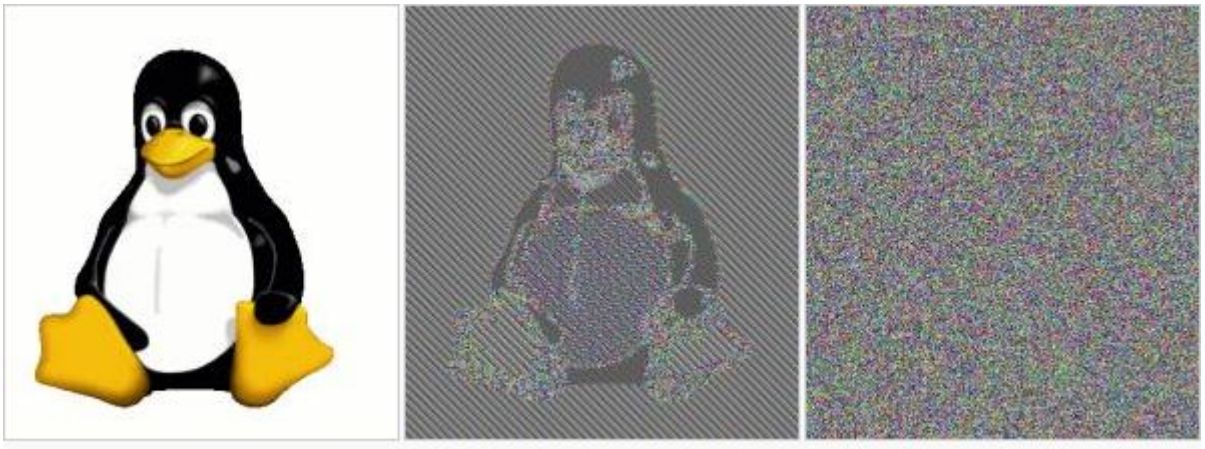
Mode	Formulas	Ciphertext
Electronic Codebook (ECB) Each block is encrypted separately from one another	Y_i=F(PlainText$_i$,Key)	Y_i
Cipher Block Chaining (CBC) Each block is first XOR'ed with previous block of ciphertext. The resulting half-baked block is then encrypted.	Y_i=PlainText$_i$ XOR Ciphertext$_{i-1}$	F(Y,Key); Ciphertext$_0$=IV
Cipher Feedback (CFB) Same as CBC, but we XOR with the ciphertext instead	Y_i=Ciphertext$_{i-1}$	Plaintext XOR F(Y,Key);Ciphertext$_0$=IV
Counter (CTR) Cipher block chaining (**CBC**) is a **mode** of operation for a block cipher (one in which a sequence of bits are **encrypted** as a single unit or block with a cipher key applied to the entire block). Cipher block chaining uses what is known as an initialization vector	Y_i=F(IV + g(i),Key);IV=token();	Plaintext XOR Y_i

(IV) of a certain length.		

The nice answer on disadvantages/advantages was given here: https://stackoverflow.com/questions/1220751/how-to-choose-an-aes-encryption-mode-cbc-ecb-ctr-ocb-cfb:

- ECB should not be used if encrypting more than one block of data with the same key.
- CBC, OFB and CFB are similar, however OFB/CFB is better because you only need encryption and not decryption, which can save code space.
- CTR is used if you want good parallelization (ie. speed), instead of CBC/OFB/CFB.
- XTS mode is the most common if you are encoding a random accessible data (like a hard disk or RAM).
- OCB is by far the best mode, as it allows encryption and authentication in a single pass. However there are patents on it in USA.

Two equal ciphertext blocks, say block 23 and block 61, have equal corresponding plaintext blocks so we can surmise that plaintext blocks 23 and 61 are identical.
There's a nice graphical demonstration of this weakness on Wikipedia, where the same (raw, uncompressed) image is encrypted using both ECB mode and a semantically secure cipher mode (such as CBC, CTR, CFB or OFB):



Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

Crypto basics 7

Comparison Chart

Confusion	Diffusion
Confusion obscures the relationship between the plaintext and ciphertext.	Diffusion spreads the plaintext statistics through the ciphertext.
A one-time pad relies entirely on confusion while a simple substitution cipher is another (weak) example of a confusion-only cryptosystem.	A double transposition is the classic example of a diffusion-only cryptosystem.
Confusion hides the relation between the ciphertext and key.	Diffusion hides the relation between the ciphertext and the plaintext.
If a single bit in the key is changed, most or all bits in the ciphertext will also be changed.	If a single symbol in the plaintext is changed, several or all symbol in the ciphertext will also be changed
In confusion, the relationship between the statistics of the cipher text and the value of the encryption key is made complex. It is achieved by substitution.	In diffusion, the statistical structure of the plain text is dissipated into long-range statistics of the cipher text This is achieved by permutation.