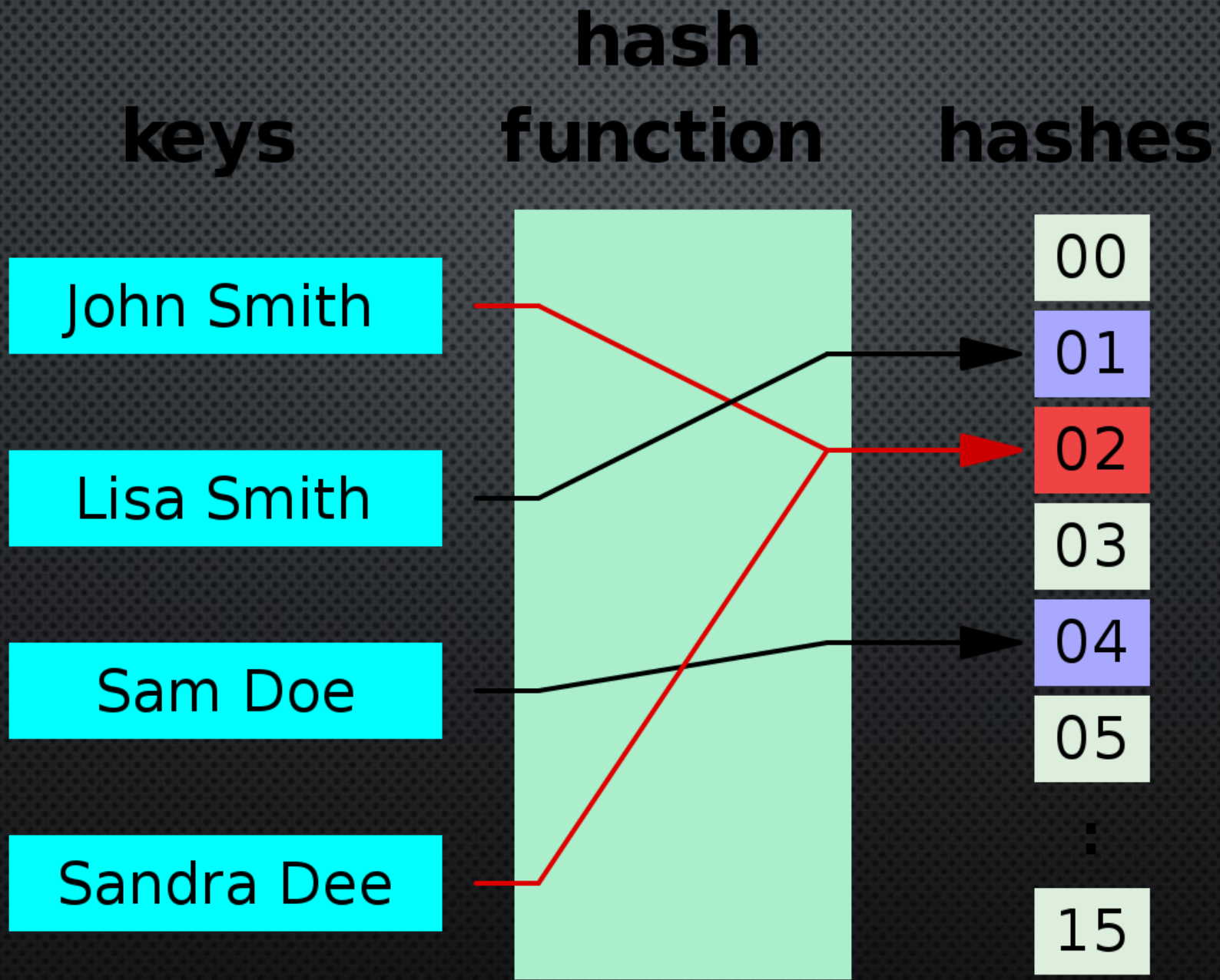
Passwords & Security

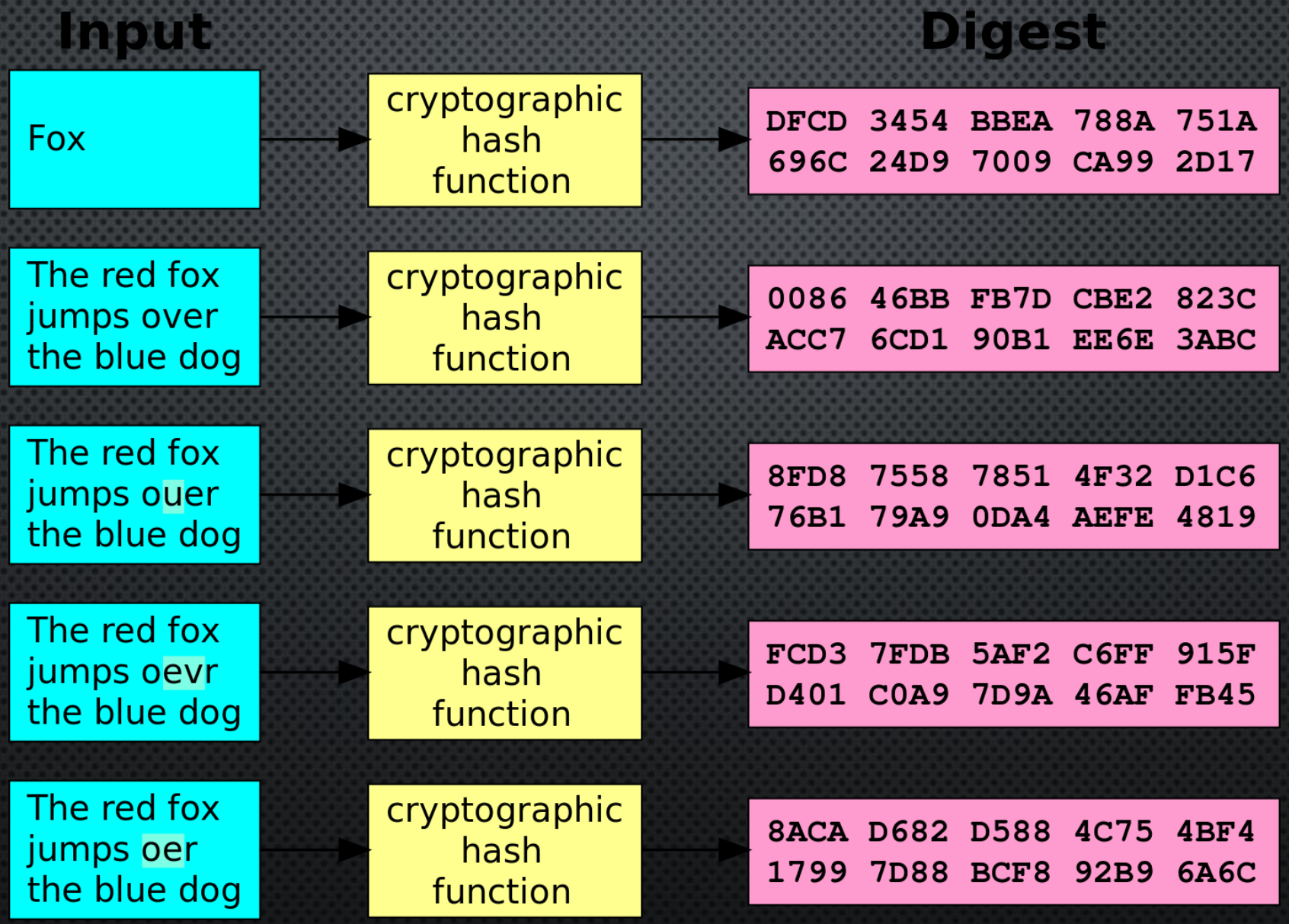
FABRIZIO_DANGELO@STUDENT.UML.EDU

TWITTER: @FABREEZ3

DISCORD: FABREEZE#2814

WHAT EVEN IS A PASSWORD??


keys
hash function
hashes
John Smith
Lisa Smith
Sam Doe
Sandra Dee
00
01
02
03
04
05
15

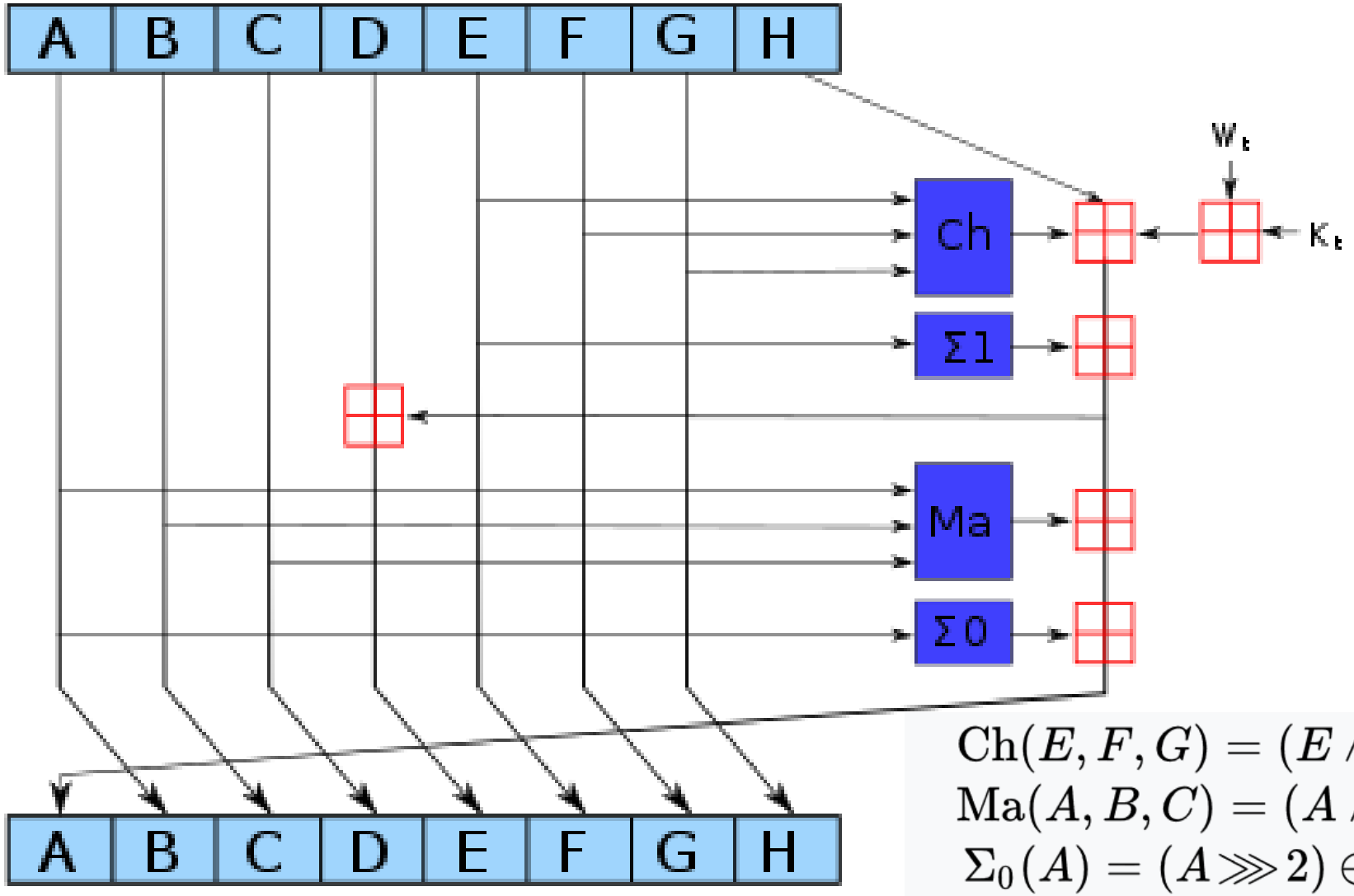

Input
Digest
Fox
cryptographic hash function
DFCD 3454 BBEA 788A 751A 696C 24D9 7009 CA99 2D17
The red fox jumps over the blue dog
cryptographic hash function
0086 46BB FB7D CBE2 823C ACC7 6CD1 90B1 EE6E 3ABC
The red fox jumps ouer the blue dog
cryptographic hash function
8FD8 7558 7851 4F32 D1C6 76B1 79A9 0DA4 AEFE 4819
The red fox jumps oevr the blue dog
cryptographic hash function
FCD3 7FDB 5AF2 C6FF 915F D401 C0A9 7D9A 46AF FB45
The red fox jumps oer the blue dog
cryptographic hash function
8ACA D682 D588 4C75 4BF4 1799 7D88 BCF8 92B9 6A6C

```
➜  ~ echo "password" | sha512sum
9151440965cf9c5e07f81eee6241c042a7b78e9bb2dd4f928a8f6da5e369cdffdd2b70c70663ee30d02115731d35f1ece5aad9b362aaa9850efa99e3d197212a  -
➜  ~ head /dev/urandom | tr -dc A-z0-9 | head -c 13; echo ''
5a]pBuDaApcuc
➜  ~ echo "password5a]pBuDaApcuc" | sha512sum
5ca731002e13b420a7b3ba6f8893b9bff56a41577a0f7da3941237094e98b16c0d8cae3017ff89ca5d0e9a421b4461e2eeee24d4e94aa66b3a657d79a5fb6870  -
➜  ~
```

```
[root@test ~]# cat /etc/shadow
root:$6$Etg2ExUZ$F9NTP7omafhKIlqaBMqng1:15651:0:99999:7:::

root:                                          -- Username
$6                                                    -- Hash
type
$Etg2ExUZ                                  -- Salt
$F9NTP7omafhKIlqaBMqng1  -- Hash
:15651                                               -- Days since
change
:0                                                    -- Days
until change
:99999                                             -- Expiration
:7                                                    --
Deactivation grace period
:::                                                  -- Account expiry
```

keyspace: 2^{256}



$$\mathrm{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$$
$$\mathrm{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$
$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$
$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

Retrieving a Hash after Windows 10 v1607

"HKLM\SAM\SAM\Domains\Account\Users\000001F4\V"

User Specific Registry Value "V"

E.g. '00000000F400000003000100F40000001A00...'

"HKLM\SAM\SAM\Domains\Account\F"

System Specific Registry Value "F"

E.g. '0300010000000000EE3DDE42E250D3010200...'

If first byte is '03' → AES Encrypted SysKey

Extracting "**Double Encrypted Hash and IV**"

Simple extraction using offsets & lengths

Length specified at "***V***" offset 0xAC (0x38 but ignored)

Hash at "***V***" offset 0xA8+0xCC

AES IV at "***V***" offset 0xB4+0xCC

3f89be20888e4878a098921d8396b535 (16 bytes)

3fd3027790ff2c0a5b8f162239d41476 (16 bytes)

Extracting "**AES Encryption Data and IV for SysKey**"

Simple extraction using offsets & lengths

Data = offset 0x88 from ***"F"*** (16 bytes)

(e.g. **7b06427ecf48cec9b61e67caed0292c9**)

IV = offset 0x78 from "***F***" (16 bytes)

(e.g. **ea322e0e26f58e4b5ab8587e75c861db**)

Decrypting "**SysKey**"

AES Decryption

With data from previous step (32 bytes)

With IV from previous step (16 bytes)

With AES Key from previous step (16 bytes)

== 32 bytes, only first 16 bytes needed

903d474b0fa91eb3003768eefcc2143d

Decrypting "**Encrypted Hash**"

AES Decryption

Where data = **Double Encrypted Hash** (first 16 bytes)

With IV from previous step (16 bytes)

With AES Key from previous step (16 bytes)

a291d14b768a6ac455a0ab9d376d8551

Constructing "**BootKey**" (16 bytes)

Concatenating in specific order

Bytes [8,5,4,2,11,9,13,3,0,6,1,12,14,10,15,7]

3f089869880dc18915d4e5adf1f6a792 →

150d8898add4f6693fc108f1a7e59289

"HKLM\System\CurrentControlSet\Control\LSA\{JD,Skew1,GBG,Data}"

System Specific Registry Classnames

E.g. JD='5d5991a3', Skew1='486c0596',

GBG='5af83341', Data='3f2cceb9'

Constructing 2 "**DES Keys**" based of RID

Concatenating and converting

Bytes [0,1,2,3,0,1,2] for DES SRC 1 (7 bytes)

Bytes [3,0,1,2,3,0,1] for DES SRC 2 (7 bytes)

→ Converted to 8 bytes with odd padding

e.g. 'f4010000' → **f40140010ea10401**

e.g. 'f4010000' → **017a01200107d002**

Decrypting "**Hash**"

DES Decryption

Where data1 & 2 = first and last 8 bytes from **Encrypted Hash**

With DES Keys from previous steps (8 bytes each)

== Two times 8 bytes to be concatenated

32ed87bdb5fdc5e9cba88547376818d4 (NTLM Hash)

OR '123456' in plain text

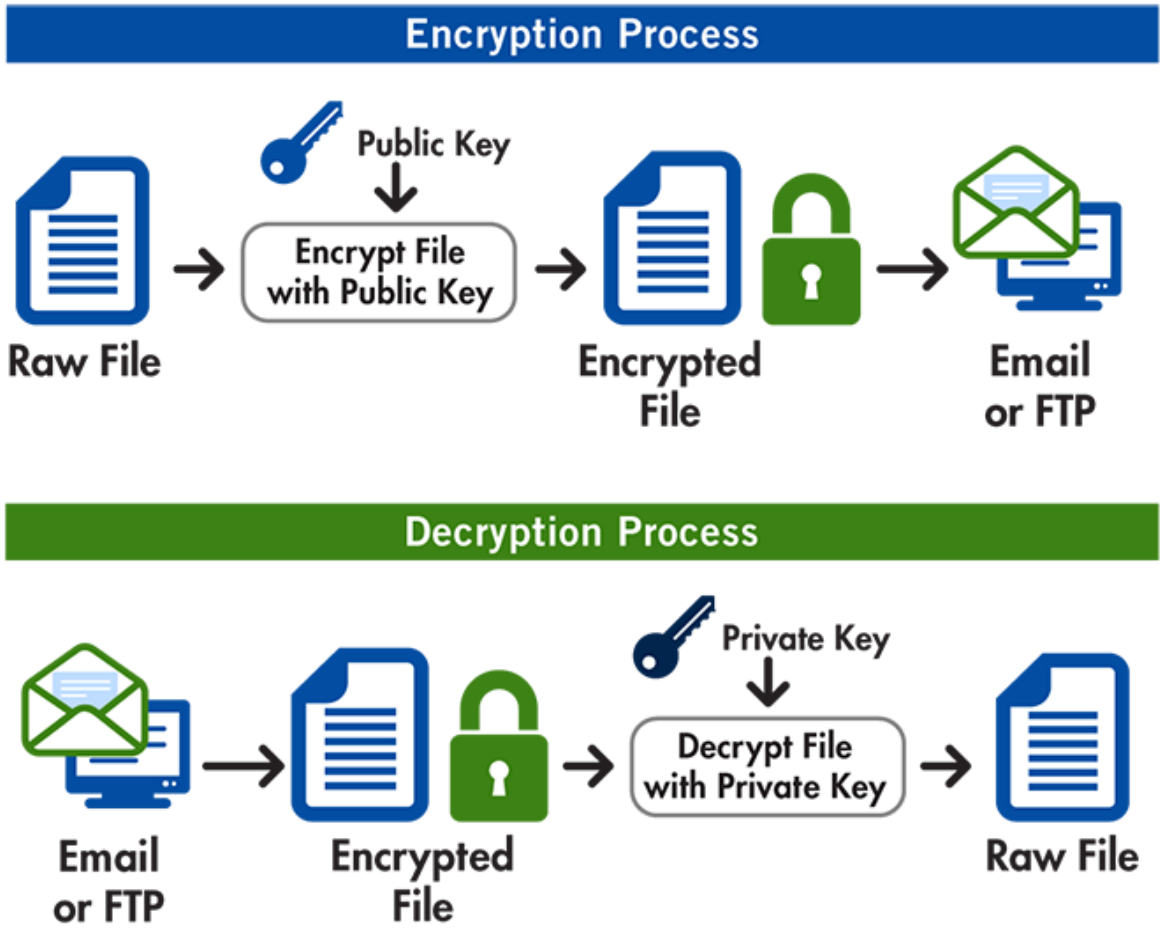
"HKLM\SAM\SAM\Domains\Account\Users\"

User RID

E.g. '000001F4' (500) reordered to 'f4010000'

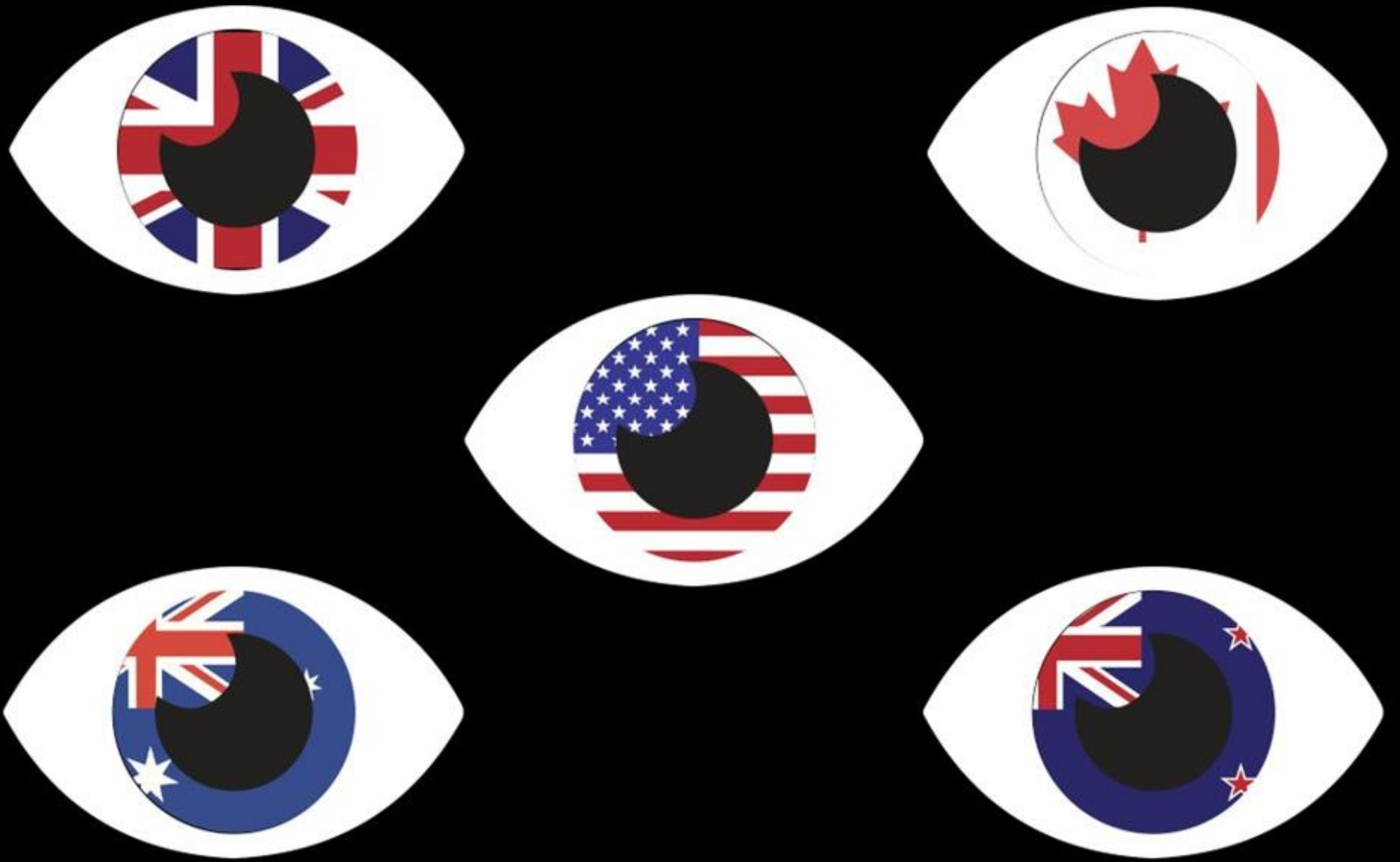
PASSWORD MANAGERS

MULTI FACTOR AUTHENTICATION

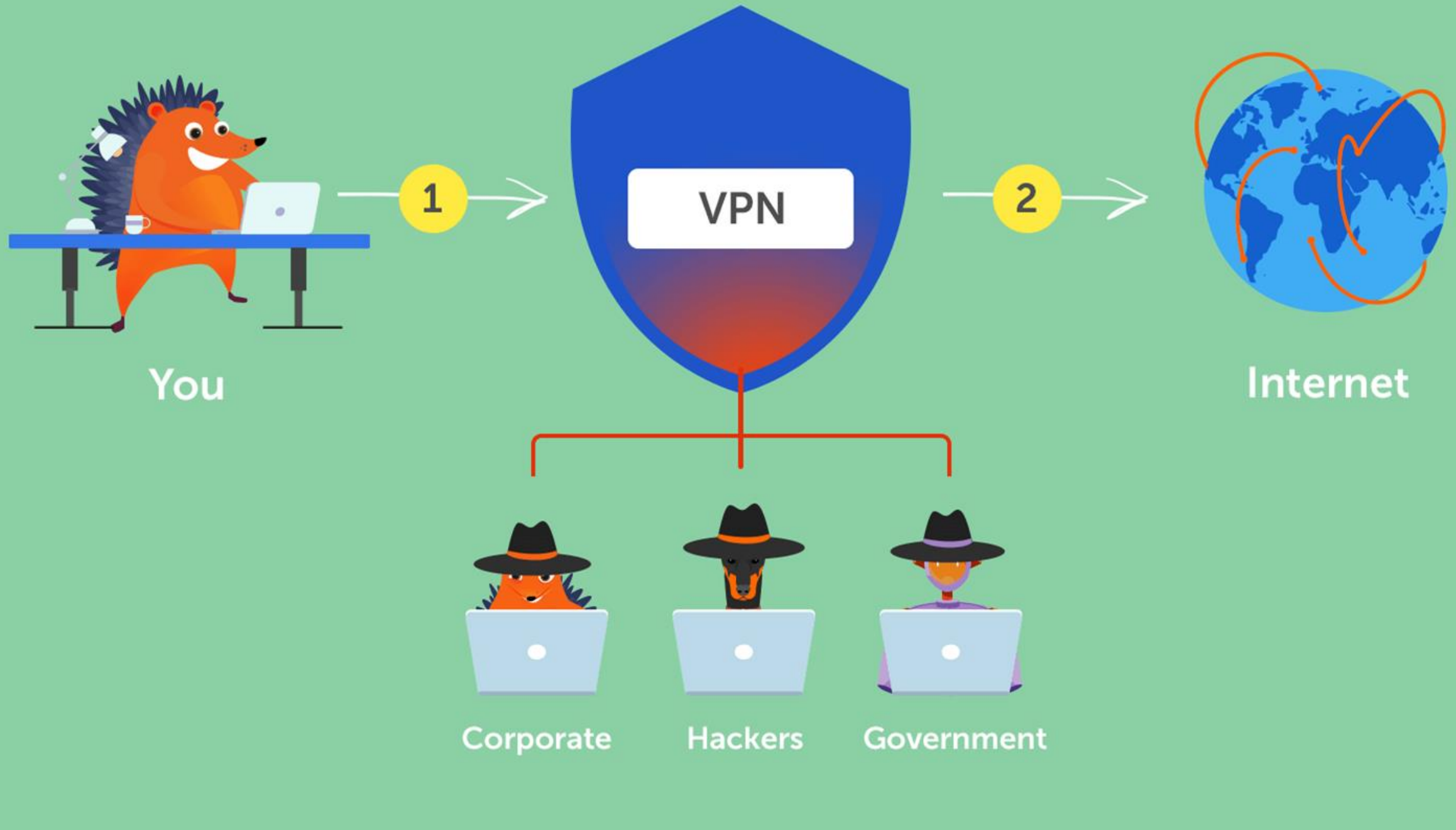


PGP ENCRYPTION

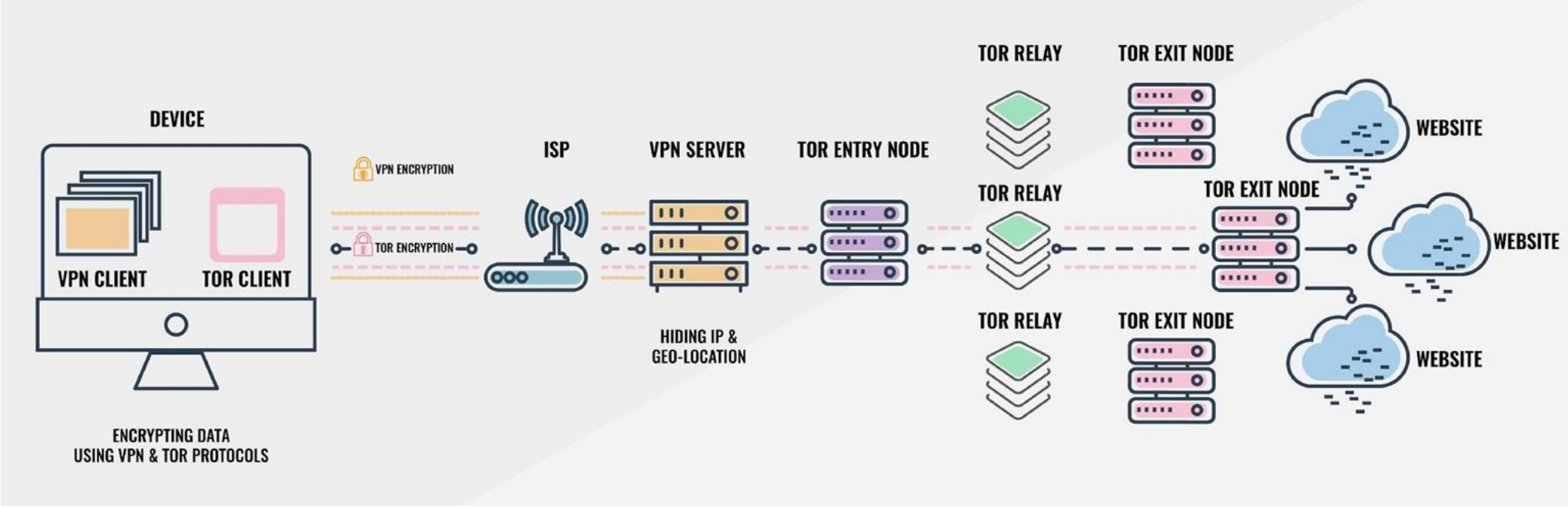
DELETING YOUR PAPER TRAIL




ProtonMail
End-to-end encrypted email, based in Switzerland.


1
VPN
2
You
Internet
Corporate
Hackers
Government


Tor Browser


DEVICE
VPN CLIENT
TOR CLIENT
ENCRYPTING DATA
USING VPN & TOR PROTOCOLS
VPN ENCRYPTION
TOR ENCRYPTION
ISP
VPN SERVER
HIDING IP &
GEO-LOCATION
TOR ENTRY NODE
TOR RELAY
TOR EXIT NODE
WEBSITE
TOR RELAY
TOR EXIT NODE
WEBSITE
TOR RELAY
TOR EXIT NODE
WEBSITE

Tails

the**amnesic**incognito**live**system