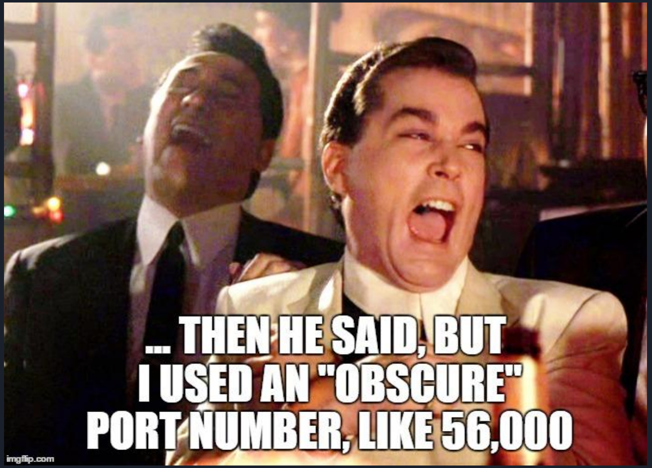
Modern Cryptography

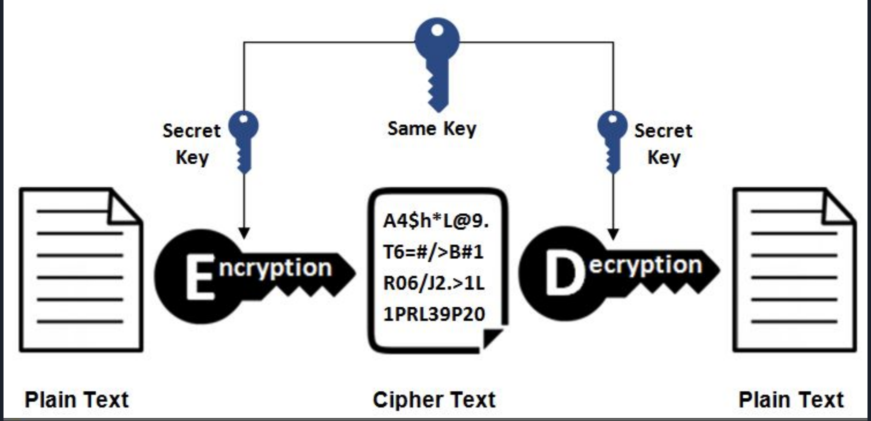
Kerckhoffs's principle

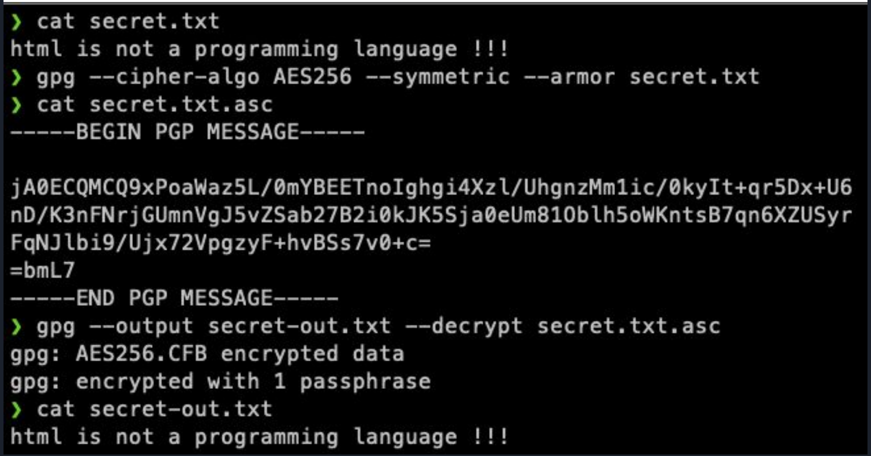


- In 1883 Dutch-born cryptographer Auguste Kerckhoffs stated multiple principes:
  - *“The security of a cryptographic system should not be based on its secrecy”*
  - “Everything about the cryptographic system should be public besides the **key**”
- This concept is widely embraced by cryptographers, in contrast to security through obscurity, which is not.

Symmetric cryptography

- Same **key** is used to encrypt and decrypt the message
- Examples
  - Vernam, DES, AES (rijndael), IDEA, …
- Can be implemented in hardware
- Fast and low resource usage
- Problem ?





```
❯ cat secret.txt
html is not a programming language !!!
❯ gpg --cipher-algo AES256 --symmetric --armor secret.txt
❯ cat secret.txt.asc
-----BEGIN PGP MESSAGE-----

jA0ECQMCQ9xPoaWaz5L/0mYBEETnoIghgi4Xzl/UhgnzMm1ic/0kyIt+qr5Dx+U6
nD/K3nFNrjGUmnVgJ5vZSab27B2i0kJK5Sja0eUm81Oblh5oWKntsB7qn6XZUSyr
FqNJlbi9/Ujx72VpgzyF+hvBSs7v0+c=
=bmL7
-----END PGP MESSAGE-----
❯ gpg --output secret-out.txt --decrypt secret.txt.asc
gpg: AES256.CFB encrypted data
gpg: encrypted with 1 passphrase
❯ cat secret-out.txt
html is not a programming language !!!
```

Symmetric cryptography: example

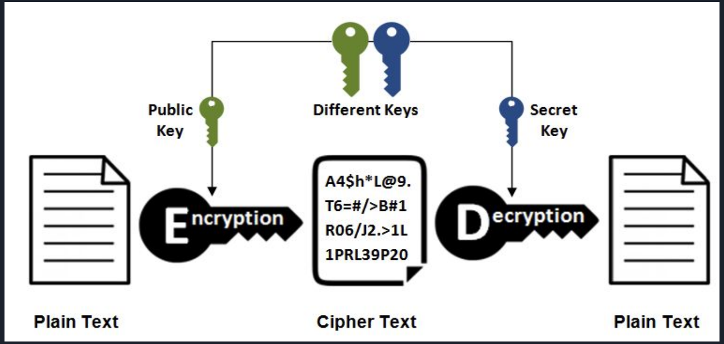
- DES : Data Encryption Standard
  - Designed and developed by IBM
  - Standard since 1977
  - Key size 56 bits
- DES is obsolete
  - 3DES is the new version
  - Key size 168 bits

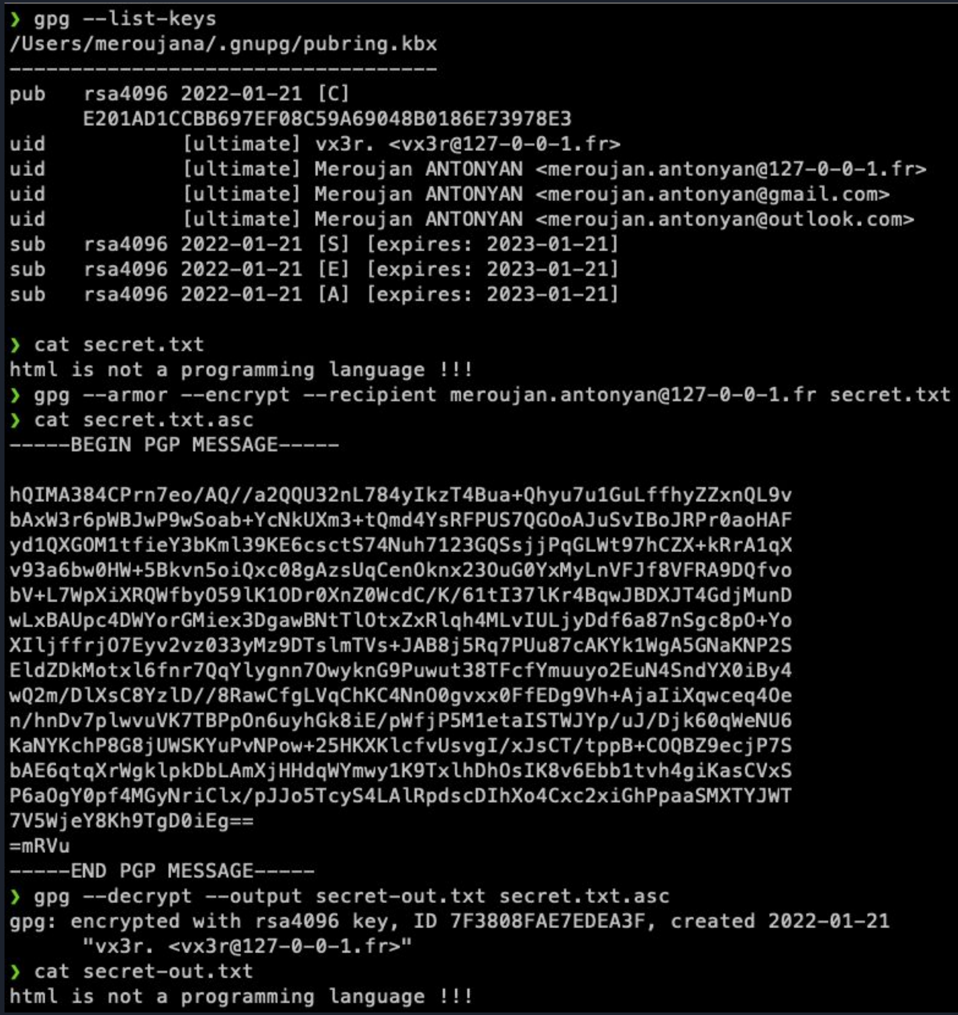
Symmetric cryptography: example

- AES : Advanced Encryption Standard (Rijndael)
    - Designed and developed by Vincent Rijmen and Joan Daemen
    - Standard since 2000
    - Selected in a competition among 20 other algorithms by NIST
    - Key size 128, 192 and 256 bits
    - De facto standard for symmetric encryption (AS OF TODAY 19/09/2022 !!!)

Asymmetric cryptography

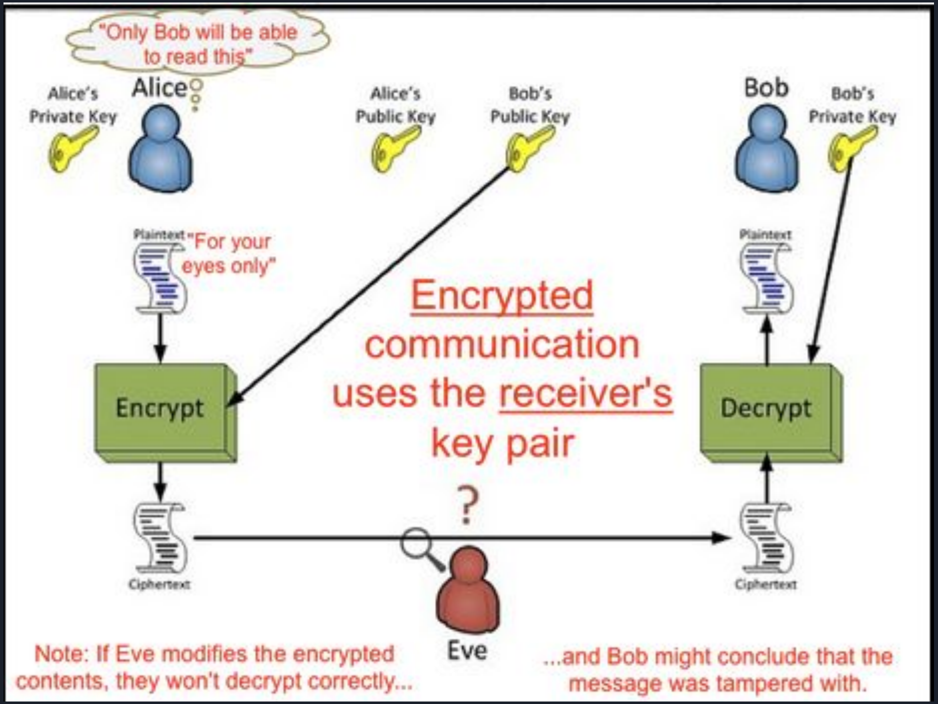
- Each actor has a pair of key (math link)
  - public : known by anyone, public info
  - private : only known by the owner, secret
- If we encrypt by one result can be decrypted only by the other !!!
- Exemples: RSA, ECDSA, DSA
- High resource, slow



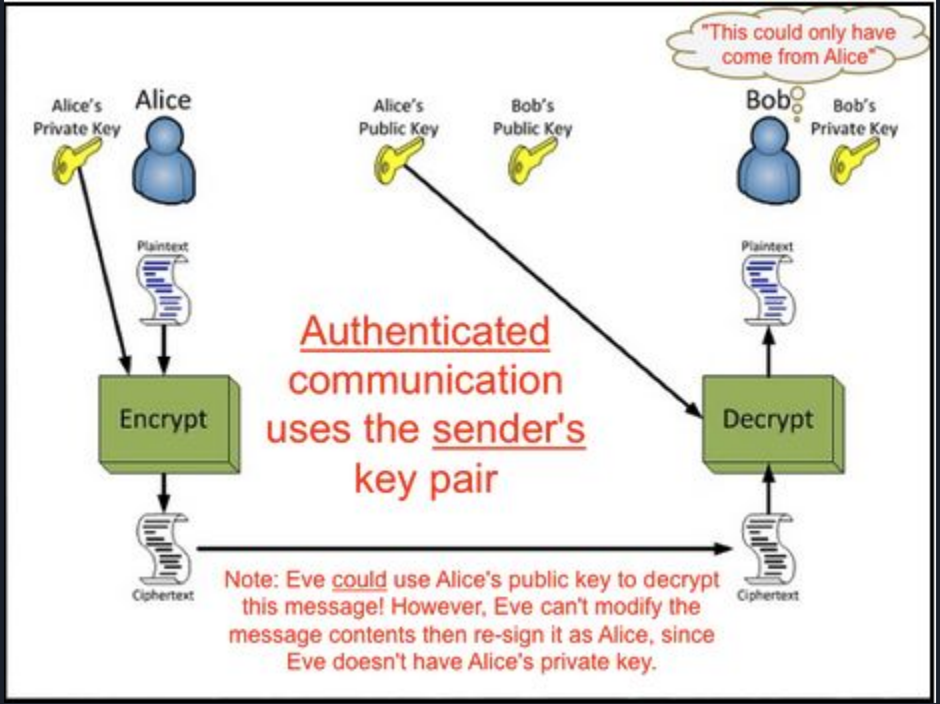


```
❯ gpg --list-keys
/Users/meroujana/.gnupg/pubring.kbx
-----------------------------------
pub   rsa4096 2022-01-21 [C]
      E201AD1CCBB697EF08C59A69048B0186E73978E3
uid           [ultimate] vx3r. <vx3r@127-0-0-1.fr>
uid           [ultimate] Meroujan ANTONYAN <meroujan.antonyan@127-0-0-1.fr>
uid           [ultimate] Meroujan ANTONYAN <meroujan.antonyan@gmail.com>
uid           [ultimate] Meroujan ANTONYAN <meroujan.antonyan@outlook.com>
sub   rsa4096 2022-01-21 [S] [expires: 2023-01-21]
sub   rsa4096 2022-01-21 [E] [expires: 2023-01-21]
sub   rsa4096 2022-01-21 [A] [expires: 2023-01-21]

❯ cat secret.txt
html is not a programming language !!!
❯ gpg --armor --encrypt --recipient meroujan.antonyan@127-0-0-1.fr secret.txt
❯ cat secret.txt.asc
-----BEGIN PGP MESSAGE-----

hQIMA384CPrn7eo/AQ//a2QQU32nL784yIkzT4Bua+Qhyu7u1GuLffhyZZxnQL9v
bAxW3r6pWBJwP9wSoab+YcNkUXm3+tQmd4YsRFPUS7QGOoAJuSvIBoJRPr0aoHAF
yd1QXGOM1tfieY3bKml39KE6csctS74Nuh7123GQSsjjPqGLWt97hCZX+kRrA1qX
v93a6bw0HW+5Bkvn5oiQxc08gAzsUqCenOknx23OuG0YxMyLnVFJf8VFRA9DQfvo
bV+L7WpXiXRQWfbyO59lK1ODr0XnZ0WcdC/K/61tI37lKr4BqwJBDXJT4GdjMunD
wLxBAUpc4DWYorGMiex3DgawBNtTlOtxZxRlqh4MLvIULjyDdf6a87nSgc8pO+Yo
XIljffrjO7Eyv2vz033yMz9DTslmTVs+JAB8j5Rq7PUu87cAKYk1WgA5GNaKNP2S
EldZDkMotxl6fnr7QqYlygnn7OwyknG9Puwut38TFcfYmuuyo2EuN4SndYX0iBy4
wQ2m/DlXsC8YzlD//8RawCfgLVqChKC4NnO0gvxx0FfEDg9Vh+AjaIiXqwceq4Oe
n/hnDv7plwvuVK7TBPpOn6uyhGk8iE/pWfjP5M1etaISTWJYp/uJ/Djk60qWeNU6
KaNYKchP8G8jUWSKYuPvNPow+25HKXKlcfvUsvgI/xJsCT/tppB+COQBZ9ecjP7S
bAE6qtqXrWgklpkDbLAmXjHHdqWYmwy1K9TxlhDhOsIK8v6Ebb1tvh4giKasCVxS
P6aOgY0pf4MGyNriClx/pJJo5TcyS4LAlRpdscDIhXo4Cxc2xiGhPpaaSMXTYJWT
7V5WjeY8Kh9TgD0iEg==
=mRVu
-----END PGP MESSAGE-----
❯ gpg --decrypt --output secret-out.txt secret.txt.asc
gpg: encrypted with rsa4096 key, ID 7F3808FAE7EDEA3F, created 2022-01-21
      "vx3r. <vx3r@127-0-0-1.fr>"
❯ cat secret-out.txt
html is not a programming language !!!
```

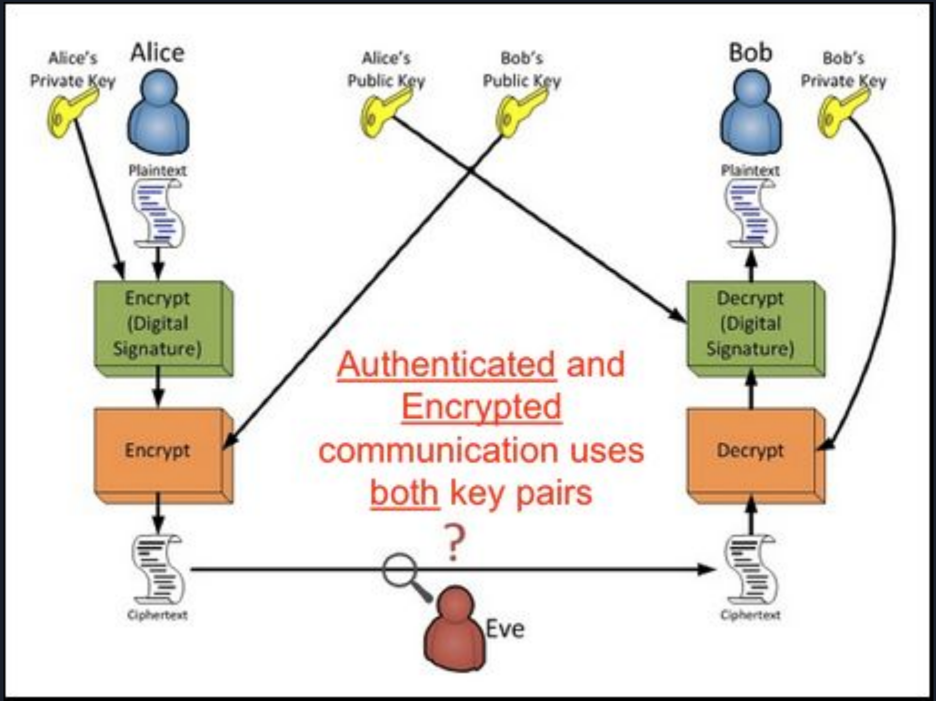
Asymmetric cryptography: confidentiality


"Only Bob will be able to read this"
Alice's Private Key
Alice
Alice's Public Key
Bob's Public Key
Bob
Bob's Private Key
Plaintext
"For your eyes only"
Plaintext
Encrypted communication uses the receiver's key pair
Encrypt
Decrypt
?
Ciphertext
Ciphertext
Eve
Note: If Eve modifies the encrypted contents, they won't decrypt correctly...
...and Bob might conclude that the message was tampered with.

Asymmetric cryptography: authentication

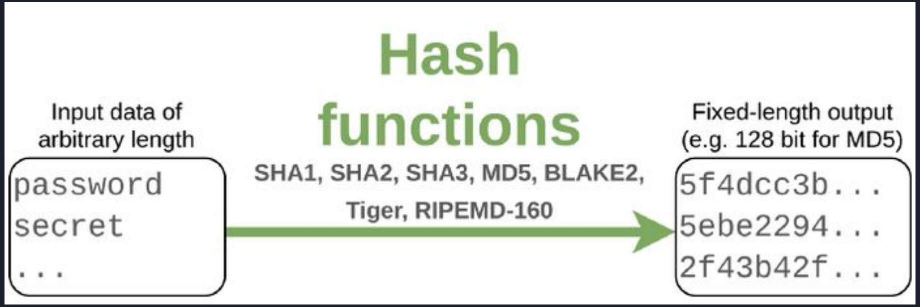

"This could only have come from Alice"
Alice's Private Key
Alice
Alice's Public Key
Bob's Public Key
Bob
Bob's Private Key
Plaintext
Plaintext
Encrypt
Decrypt
Authenticated communication uses the sender's key pair
Ciphertext
Ciphertext
Note: Eve could use Alice's public key to decrypt this message! However, Eve can't modify the message contents then re-sign it as Alice, since Eve doesn't have Alice's private key.

Asymmetric cryptography: confidentiality and authentication

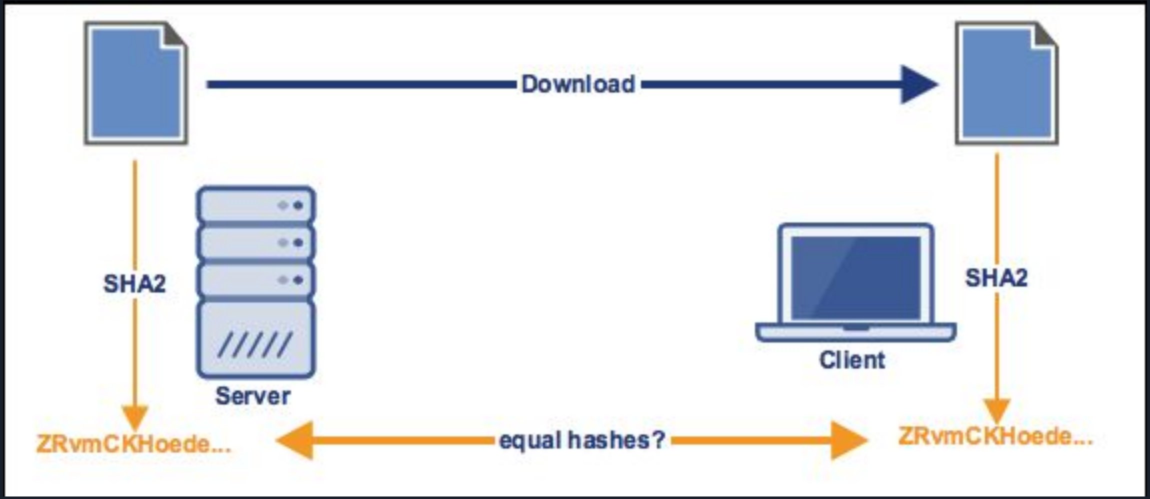


Hashing fonction

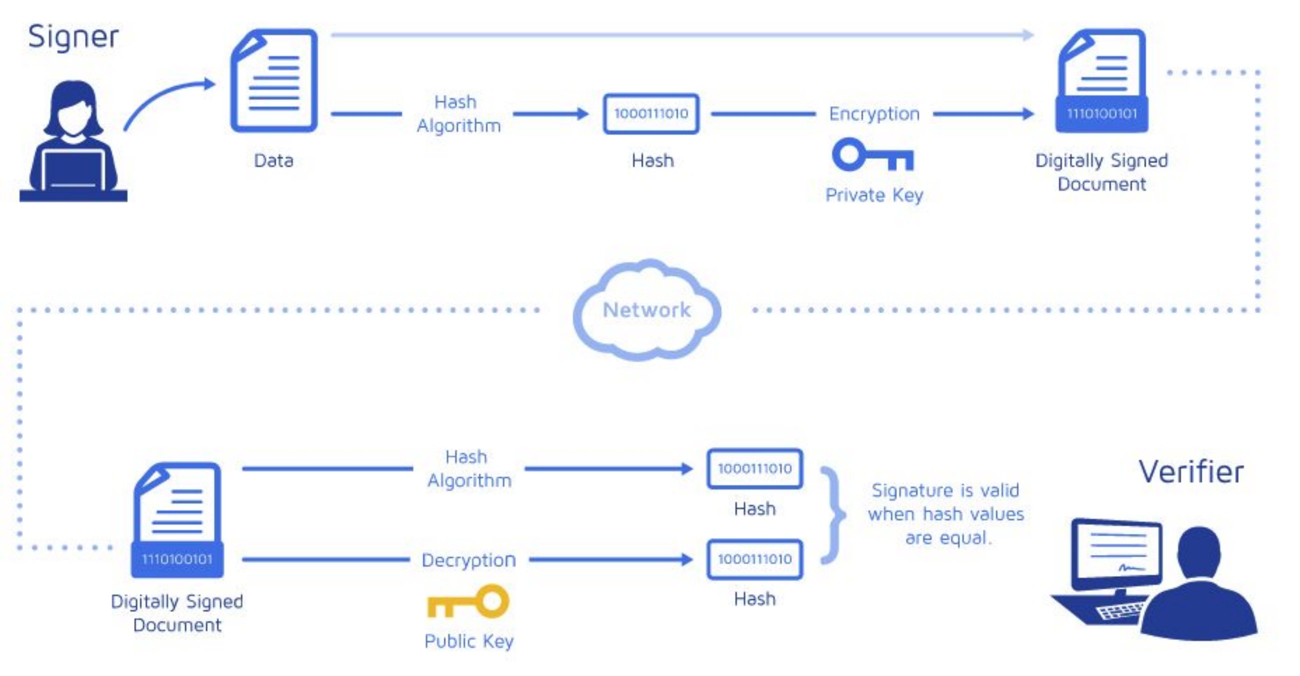
- One way operation
- Fixed lengths output
- Unique fingerprint of data
- Examples
  - MD5 : Message Digest 5
    - Output 128 bits digest
  - SHA-X : Secure Hash Algorithm
    - Output from 160 to 512 bits
  - RIPEMD-160
    - Output 160 bits digest


Hash functions
Input data of arbitrary length
password
secret
...
SHA1, SHA2, SHA3, MD5, BLAKE2, Tiger, RIPEMD-160
Fixed-length output (e.g. 128 bit for MD5)
5f4dcc3b...
5ebe2294...
2f43b42f...

Hashing fonction: usage



Hashing fonction: digital signature



GPG / PGP

- Generate a key pair
- Encrypt, sign and send out
- Receive, verify the signature and decrypt
- How ? RTFM