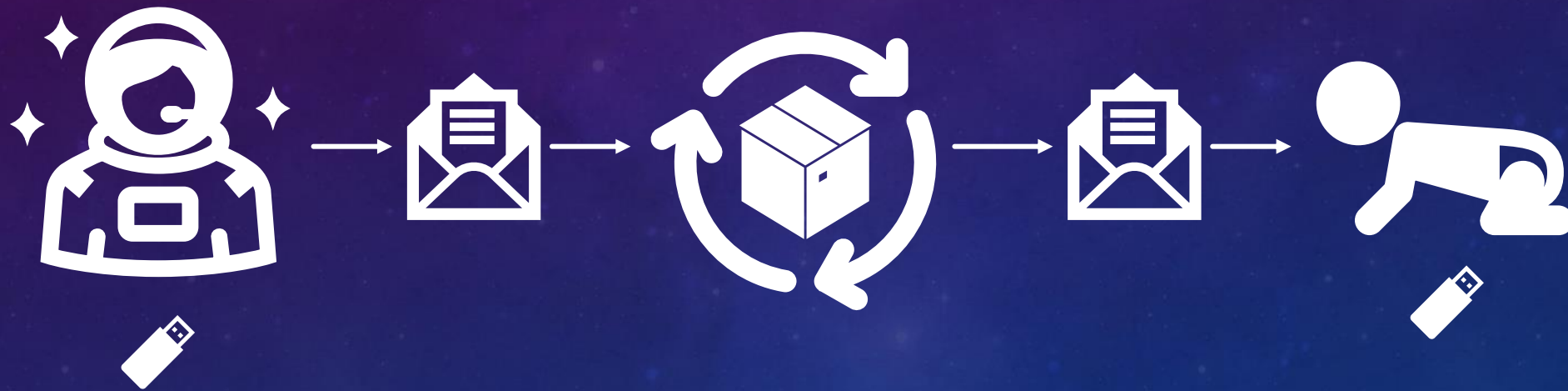


The background is a dark blue gradient with a subtle pattern of white dots. On the left side, there are several concentric circles and arcs. One large arc features a scale with numerical labels: 140, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, and 260. Other smaller circles and arcs are scattered across the left and top portions of the image, some with arrows indicating a clockwise direction.

CRYPTANALYSE

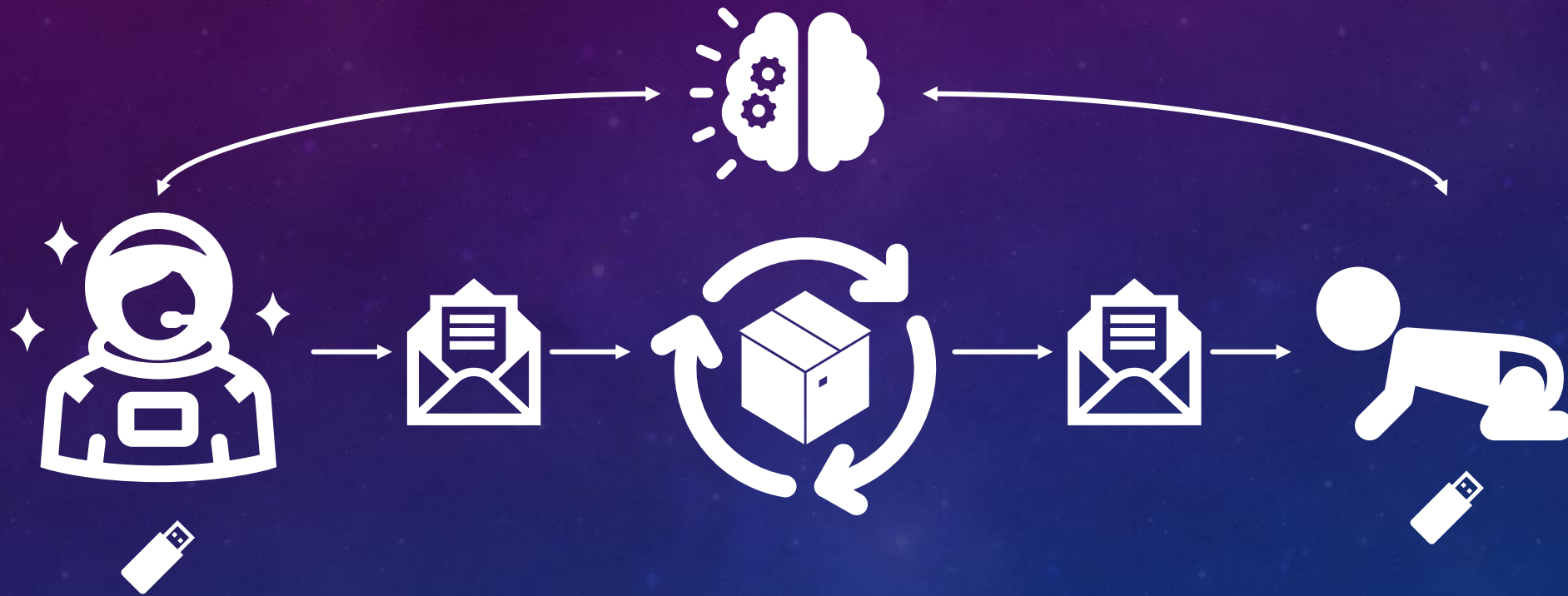
COMMENT CASSER LE CHIFFREMENT RSA ?

CRYPTOGRAPHIE SYMÉTRIQUE



Chiffrement de Vigenère, Machine Enigma, Chiffrement AES...

CRYPTOGRAPHIE ASYMÉTRIQUE



Chiffrement RSA, Problème du Logarithme Discret, Chiffrement d'ElGamal...

CHIFFREMENT RSA

On choisit p et q (Premiers) tel que :

$$N = p \times q$$

On calcule :

$$\varphi(N) = (p - 1)(q - 1)$$

On choisit e tel que :

$$\text{PGCD}(e, \varphi(N)) = 1 \quad \Leftrightarrow \quad d.e - k.\varphi(N) = 1$$

On a donc :

Clé Publique : $f(N, e)$

Clé Privée : $f(N, d)$

CHIFFREMENT RSA

*Clé Publique : $f(N, e)$
Clé Privée : $f(N, d)$*

$$\begin{aligned} N &= p \times q \\ \varphi(N) &= (p - 1)(q - 1) \\ d.e - k.\varphi(N) &= 1 \end{aligned}$$



Le Bébé chiffre le message :

$$C \equiv M^e \bmod N$$



L'Astronaute déchiffre le message :

$$C^d \equiv M \bmod N$$

$$C^d \equiv (M^e)^d \equiv M^{e.d} \equiv M^{1+k.\varphi(N)} \equiv M \bmod N$$

UTILISER OPENSSL

Extraire le Module N et l'exposant e de la clé public :

```
openssl rsa -in public.pem -pubin -text -modulus
```

Générer des clés :

```
openssl genrsa -out private.pem 2048  
openssl rsa -in private.pem -pubout -out public.pem
```

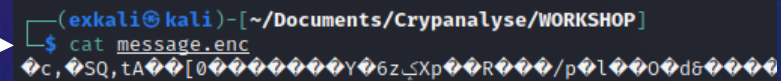
```
(exkali@kali) - [~/Documents/Crypanalyse/WORKSHOP]
$ cat public.pem
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA3gKF4SkENJTLgFD4Nr75
9WgPOwTr/wfC8w+Y1VCMbxxRXZ5T6KBDzXZKEnKAcSLaLkP8Hxb/j6m1g0sa7QZ
QhcT0JM+S+Ga0LG8zWhdTN2rl3Hm+Wh3W7/53W7r/8wn40L6IMroksM0TAFFR5bd
uJw1r4Xw0B616X8pQSmWi5I+epMR97d77ShxU2p1wyAKVEZ7dFDTqoph2PhjI/ZT
bqu7XV/0enAM/F+28/CbRkRovP2D519QVcnFMBQCBM+Kt2Xd/aFSL66yFDAJ9n
aTp3JfdiBzPCikBJ0fRRY/bhep064yZMz5WsdIQvbnF5yDEb77i+sdpfUusgJj
YQIDAQAB
-----END PUBLIC KEY-----

(exkali@kali) - [~/Documents/Crypanalyse/WORKSHOP]
$ cat keyTest1.pem
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA3gKF4SkENJTLgFD4Nr759WgPOwTr/wfC8w+Y1VCMbxxRXZ5
T6KBDzXZKEnKAcSLaLkP8Hxb/j6m1g0sa7QZQhcT0JM+S+Ga0LG8zWhdTN2rl3Hm
+Wh3W7/53W7r/8wn40L6IMroksM0TAFFR5bdUJw1r4Xw0B616X8pQSmWi5I+epMR
97d77ShxU2p1wyAKVEZ7dFDTqoph2PhjI/ZTbqu7XV/0enAM/F+28/CbRkRovP2D
519QVcnFMBQCBM+Kt2Xd/aFSL66yFDAJ9naTp3JfdiBzPCikBJ0fRRY/bhep06
4yZMz5WsdIQvbnF5yDEb77i+sdpfUusgJjYQIDAQABAOIBAfmZxxKyjSHznHUt
feAKf7NNxifZuq15InMarmIZWjRVydsfrz2rMPIdx+weqrWfVwC0DR0Abynj7t
yJ7HVxJXyGiB1h0uxjTQxHbtwYsvIyZDfws4tyFZQK+nCIoueec7jca7Uf13F9
acqU1vN1CC3I7sSXkG+mhk3p7Ppy0J3vkNAJu+8sh1AqPw30LAi90nsSDDLQgWE
0l3JNMU8g27hfvgBs80+7liBfr4XJ77hDcRgq462YA0JwKLB/iARDV11dasxRcDs
AWdZyWE1xUsC03WUkronXnHaJ6qqtkbYi/HpLmzeK2LewOIFLbnFluGfY0Xw9+Pw
vQy2QUkCgYEA/+ME1WUv6j4sd87csV82Sg5T0yVGBhDNbea8KUMz9hoEAVd+sSh
Pwke1SUwi3RQUJjmsz1jj/56uCaCgslpa5xYq/Atirf51nX0sMxVPUf0YeAuIAP4
3JiESSXu2JjvA3cA7TsTuYz8u9ygZZP7jioDa/g46Z5sRtCr7j7LrmcCgYEA3huq
0eu4n7mVfClGdf3qfIc4tIbu17KsYE2aADMut+fVZfahp1j0BaoFjE6IxYgIjL
KIw3E8Ht6KG35wqbBmUaRGH0AZ/Lu2fNaEus5sHs3LjxA15YLSxLShoZtHG7YKQr
1HTHwY7v1FJLQhWP2IXjhHYu+0HT0PtHvKRMvcCgYA2c57BsypfpptVboV+BNXu
chMA9vD7M26LvchwoRQTP/SWPiIvPL7MPuoq8NKyboVxkU2arUtnTmMGLtP0YBn
b9vGgtNmI4Q8yND0+mZhJf1bB10JaI76kLIEJth0qyyhdh9yXBH0eDYH3Akdl1HB
XNthjG6/+ABeiUusy6GgnQKBgEFcB9ZuJDKFln85q8UhlC4hI+QnIVZPMUa6hl
04pQXEIHdtF+JLVXz8j9T83n3rkleEvZpUvLFAEVDYwaSZf3m/pOACiMqX1wey7
D3pd0HKqBGaw+cGtUa1FIhoQL4K3rX1+htHBq0eyIHo9RCwq7Q2RL0c69ejFVh
4DFRAoGBAPgH6gKMQEdx0TYA3AJtWwxCOEM05EMUoa9Qd2erg5W3GGx3XUa89PQD
ZwifaR9PcPqm8X86KI7BWuouYncBqz4qA1EHNEqZgHhCquYmvV7tFGsmcyj/rEr
UPqXpLA6rVN8cwXBfZgbSaRkrMj6n7jKwCmG1UED03sxhJz7NFxi
-----END RSA PRIVATE KEY-----
```

UTILISER OPENSSL

Encrypter un message à partir de la clé public de l'Astronaute (Bébé):

```
openssl rsautl -encrypt -inkey public.pem -pubin -in message.txt -out message.enc
```



```
(exkali@kali)-[~/Documents/Crypanalyse/WORKSHOP]  
$ cat message.enc  
c,SQ,tA[0000000000Y6z_XpR/p1000d60000
```

Décrypter un message à de la clé privé de l'Astronaute (Astronaute) :

```
openssl rsautl -decrypt -inkey private.pem -in message.enc -out recover.txt
```

A VOUS DE JOUER



Vous êtes à la place de l'Astronaute. Vous avez généré une clé Privée puis une clé publique. Le Bébé vient de vous envoyer un Message encrypté avec votre clé publique.
A vous de retrouver le message.

```
openssl rsautl -decrypt -inkey private.pem -in message.enc -out recover.txt
```

```
Ceci un petit test pour vérifier si openssl fonctionne bien.  
Pour plus d'informations sur la cryptographie : vous pouvez aller voir ce site :  
http://www.5z8.info/10101110010110101001\_t3c4wr\_getPersonalData-start
```


CRYPTANALYSE



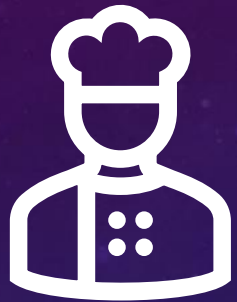
- Factorisation du module N
- Théorème des Restes Chinois : Un message, Plusieurs Destinataires
- Algorithme d'Euclide Etendu : Plusieurs Messages, Modules Communs (N)
- D'autres Algorithmes que je ne comprends pas...
 - Attaque de Weiner
 - Attaque de Coppersmith
 - Attaque de Bleichenbacher
 - Attaque par Exposition de Clé (Corruption de Clé)

LES OUTILS



- OpenSSL : Decrypt / Encrypt / Construction de Clé
- <http://factordb.com/> : Trouver des Factorisations d'Entiers
- <https://www.dcode.fr/base-64-encoding> : Base64 \Rightarrow Integer / Hexa
- <https://www.rapidtables.com/convert/number/decimal-to-hex.html> : Big Integer \Rightarrow Hexa
- CyberChef : Hexa \Rightarrow Base64
- <https://superdry.apphb.com/tools/online-rsa-key-converter> : Construire Public.pem

A VOUS DE JOUER



Module N :

19075564050606964910614504326460288610811
79759533184460647975622318915025587184175
75405497615512159329349226046415263009323
85092466032074171247261215808581859859389
46945490481721756401423481

Exposant e :

65537

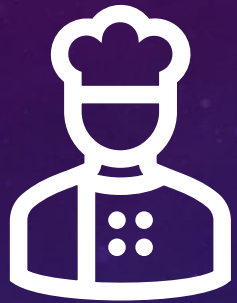
Message Chiffré C :

AUvouDJxQNTdZaNp880Fqp8emFbdY+EkE5TqKy0gj
dIK4wPwpJ8agp9Ltya8Ypkn75DGhNZYQ+QC
WUtZDwKQXq4mZMjJiDV1TxVkRSZucw==

A VOUS DE JOUER

Reconstruire la clé privée

Calculs à effectuer :



p, q, e

$$\varphi(N) = (p - 1)(q - 1)$$

$$d \equiv e^{-1} \text{ mod } \varphi(N)$$

$$e_1 \equiv d \text{ mod } (p - 1)$$

$$e_2 \equiv d \text{ mod } (q - 1)$$

$$\text{coefficient} \equiv q^{-1} \text{ mod } p$$

Construction de la clé :

Remplir le fichier Template

```
openssl asn1parse -genconf constructkey.txt -out newkey.der  
openssl rsa -inform DER -outform PEM -in newkey.der -out privatekey.pem
```


BONUS



Factoriser :

251959084756578934940271832400483985714292821262040320277771
378360436620207075955562640185258807844069182906412495150821
892985591491761845028084891200728449926873928072877767359714
183472702618963750149718246911650776133798590957000973304597
488084284017974291006424586918171951187461215151726546322822
168699875491824224336372590851418654620435767984233871847744
479207399342365848238242811981638150106748104516603773060562
016196762561338441436038339044149526344321901146575444541784
240209246165157233507787077498171257724679629263863563732899
121548314381678998850404453640235273819513786365643912120103
97122822120720357

RECOMPENSE DE 200 000 \$ A LA CLE*

LIENS INTÉRESSANTS

<https://opensource.com/article/21/4/encryption-decryption-openssl>

COMMENT SE FAIT L'ÉCHANGE DE CLÉS ?

Principe d'échange de clé par la méthode Diffie-Hellman.
Peut-être pour un autre Workshop ?