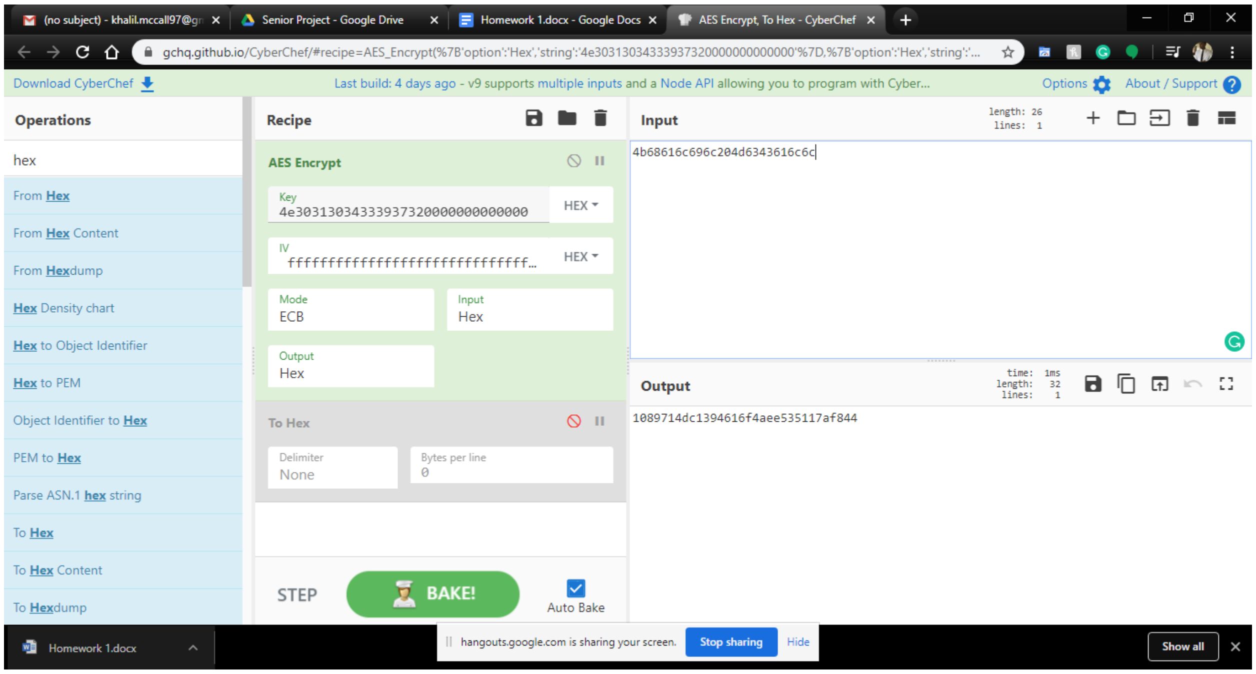
1. [2]

Use Cyberchef ([https://gchq.github.io/CyberChef/ (Links to an external site.)](https://gchq.github.io/CyberChef/)) for following questions:
a) Use AES to encrypt “Your Name” with your N# as the key parameter. (pad it with zeros).
Select ECB mode, and set IV = ffffffffffffffffffffffffffffffff
Enter Input, Key, and Output in Hex format.
Deliverable(Snapshot)



gchq.github.io/CyberChef/#recipe=AES_Encrypt(%7B'option':'Hex','string':'4e303130343339373200000000000000'%7D,%7B'option':'Hex','string':'...

Last build: 4 days ago - v9 supports multiple inputs and a Node API allowing you to program with Cyber...

Recipe

AES Encrypt

Key: 4e303130343339373200000000000000 HEX

IV: ffffffffffffffffffffffffffff... HEX

Mode: ECB

Input: Hex

Output: Hex

To Hex

Delimiter: None

Bytes per line: 0

Input

length: 26
lines: 1

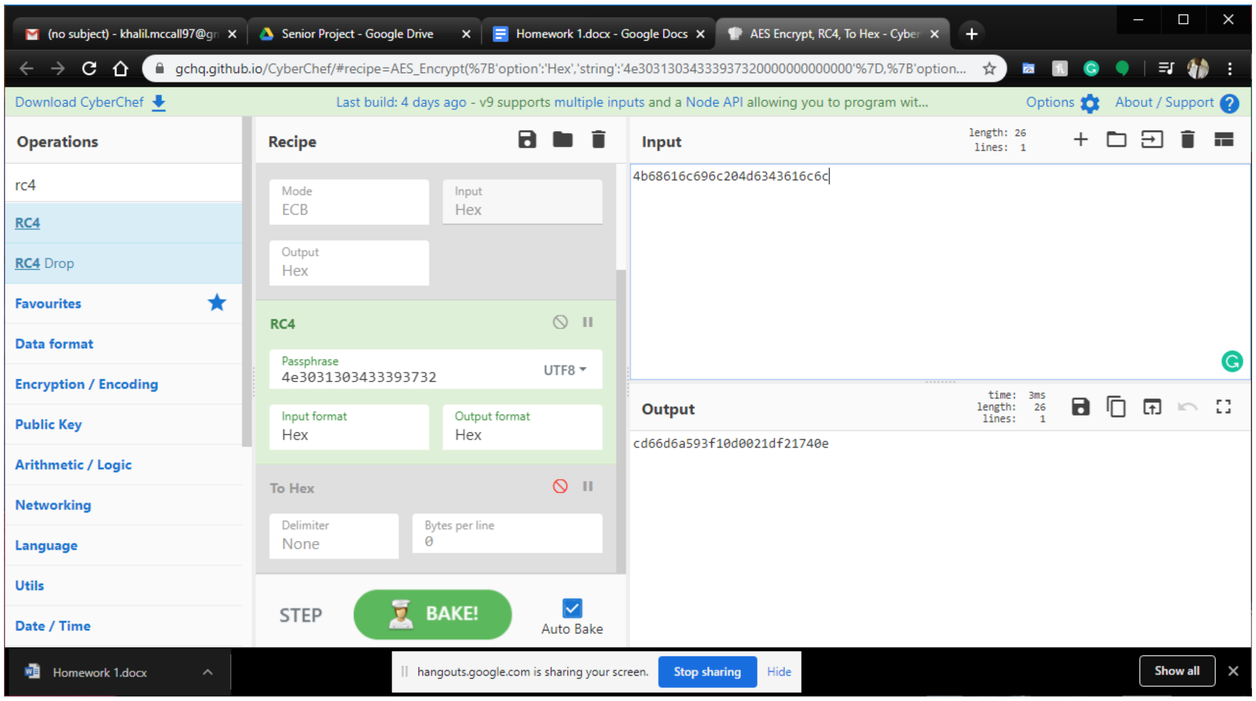
```
4b68616c696c204d6343616c6c
```

Output

time: 1ms
length: 32
lines: 1

```
1089714dc1394616f4aee535117af844
```

b) Use RC4 to encrypt “Your Name” with your N# as the key parameter.
Enter Input, Key, and Output in Hex format.
Deliverable(Snapshot)


(no subject) - khalil.mccall97@g
Senior Project - Google Drive
Homework 1.docx - Google Docs
AES Encrypt, RC4, To Hex - Cyber
gchq.github.io/CyberChef/#recipe=AES_Encrypt(%7B'option':'Hex','string':'4e30313034333937320000000000000'%7D,%7B'option...
Download CyberChef
Last build: 4 days ago - v9 supports multiple inputs and a Node API allowing you to program wit...
Options
About / Support
Operations
rc4
RC4
RC4 Drop
Favourites
Data format
Encryption / Encoding
Public Key
Arithmetic / Logic
Networking
Language
Utils
Date / Time
Recipe
STEP
BAKE!
Auto Bake
Homework 1.docx
hangouts.google.com is sharing your screen.
Stop sharing
Hide
Show all

```
Mode: ECB    Input: Hex
Output: Hex

RC4
Passphrase: 4e3031303433393732   UTF8
Input format: Hex    Output format: Hex

To Hex
Delimiter: None    Bytes per line: 0

Input    length: 26  lines: 1
4b68616c696c204d6343616c6c

Output   time: 3ms  length: 26  lines: 1
cd66d6a593f10d0021df21740e
```

2. [2]
What are the public keys and public key algorithms of www.google.com and www.twitter.com ?

Google:

-Public Key: 256 bytes : B4 A9 74 73 30 65 80 0D 4A B4 55 4C 98 79 74 F5 07 1B A2 9E 92 25 EC FF 13 58 C0 40 30 F8 3D E8 C9 EF 90 9B 6B F3 74 9E FC 12 0B 39 22 BD 66 31 59 01 0A 01 56 92 1B C8 AA CD 48 95 F2 E2 99 91 79 BA 62 F7 3E 91 D9 DD DC F0 19 8D 2B 98 28 99 E6 7D 46 8D 72 7E 10 A0 DA 2F 8A 28 4B A8 75 8F F4 B3 7F 6B C1 F6 55 2B F1 E2 E6 6F 15 99 B3 D0 06 FE 49 02 E2 A6 D5 23 43 88 E1 3A 4F 25 FD 5F F8 0B E8 9E 12 84 C7 FC 3E A3 2E C6 1F 5C 65 F9 9C 3B BD 04 F4 DB 55 4E DD 56 C6 FF C3 42 29 6F 94 24 22 03 EA 7A CF AB 0F E8 A1 64 BF 58 19 14 EE 1D CC 37 DF C7 A6 39 AF CC F3 68 06 F3 22 70 80 85 8A 3D 48 B6 3C 84 9D BD 06 18 01 0B 97 AD BA 28 73 20 F6 9F D3 73 F7 97 49 C3 E8 DA 26 74 A2 1E 4F C5 25 A0 60 37 C4 C4 16 50 7A DD 25 59 C8 55 A7 84 67 23 F0 87 24 14 1E EB 53 1B D0 99

-Public Key Algorithm:

RSA Encryption (1.2.840.113549.1.1.1)

Twitter:

-Public Key: 256 bytes : E6 57 DA 47 25 B5 FA DC 3D 3C 9F 00 01 6D 20 08 13 B9 E8 80 A9 E5 3F 93 A3 37 38 0A EB 39 34 49 18 BB 8B 0A CB E3 DD AF 8E D9 8E 1C C4 1B CF CA 1B 00 81 B3 3E 9C B9 57 B5 FD 33 88 7E 52 0E 32 73 2C EE A6 54 AE 93 EF 5C 59 3A 32 3C CF 4D 47 56 46 F0 A8 E9 C5 54 63 C3 F3 65 F2 81 7E 16 D6 86 A3 3A DE 1D D7 03 29 39 9A 1C E8 1F CB 87 EC BB 40 21 54 BC CF B1 74 C0 F4 F3 92 72 AD 66 6F 68 6C 37 A1 04 2A E0 36 EB 0C 16 A8 58 26 D2 CD D6 DB B9 19 35 C6 98 1C B4 DD B1 77 9A C5 FE 7E 4C 83 85 24 18 1C 93 47 F3 44 7C 1F 65 B9 58 A8 F9 B6 D3 A3 8B 4F 88 A4 5B C0 ED A7 CE 81 86 58 C6 92 F1 3F 94 12 D4 E9 7A 5D D8 5C FA 54 B0 FD 9F 91 C3 C5 CE 98 6D E9 E6 2B 3A 2E EA 86 D6 AE 81 6F 29 7A CD E3 C8 F8 71 C6 9F 77 B6 F3 47 D8 EA FB 49 A0 60 E9 C3 3A 98 48 88 8C DD 84 CF CB

-Public Key Algorithm: RSA Encryption (1.2.840.113549.1.1.1)

What signature algorithms are applied for their digital certificate?
Deliverable(Text)

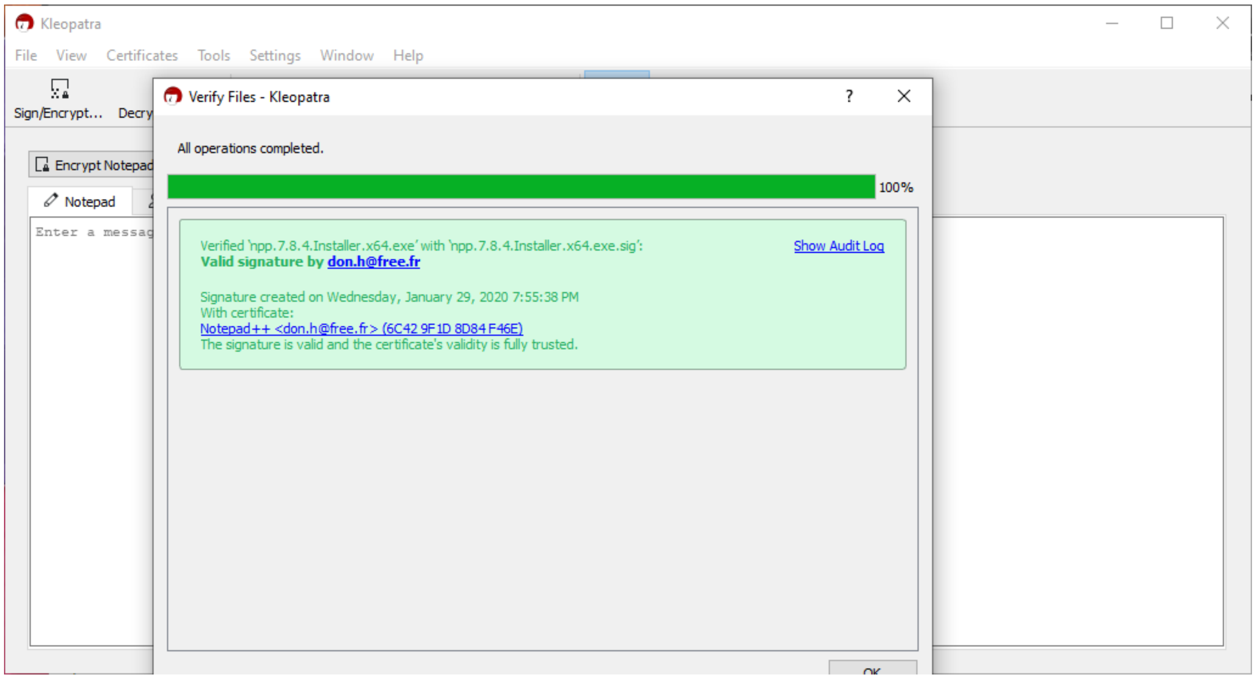
Google: SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)

Twitter: SHA-256 with RSA Encryption (1.2.840.113549.1.1.11)
3. [2]
Check the following AES encryption example:
[https://howtodoinjava.com/security/java-aes-encryption-example/ (Links to an external site.)](https://howtodoinjava.com/security/java-aes-encryption-example/)
What is the final key value in this example, explain your answer?
Deliverable(value of the key in Hex format and explanation)

a. 4e 31 36 75 32 58 59 71 32 63 2f 71 33 69 6a 5a 79 35 39 76 49 67 3d 3d
b. The originalstring and secret key are passed into the encrypt function
    - Secret key passed into setkey() function
    - Key is turned into utf8
    - key is hashed into message digest algorithm “SHA-1”
    - Key is ensured 16 blocks
    - new instance of secret key is created
    - new instance of cipher created and ciphered in encrypt mode.
    - string returned
c. For decrypt:
    - the encrypted string and the secret key are passed into the decrypt function
    - Secret key passed into setkey() function
    - Key is turned into utf8
    - key is hashed into message digest algorithm “SHA-1”

- Key is ensured 16 blocks
- new instance of cipher created and is ciphered in decrypt mode
- string returned

4. [2]
Download Notepad++ installation file and verify it’s GPG signature.
Deliverable(Snapshot)

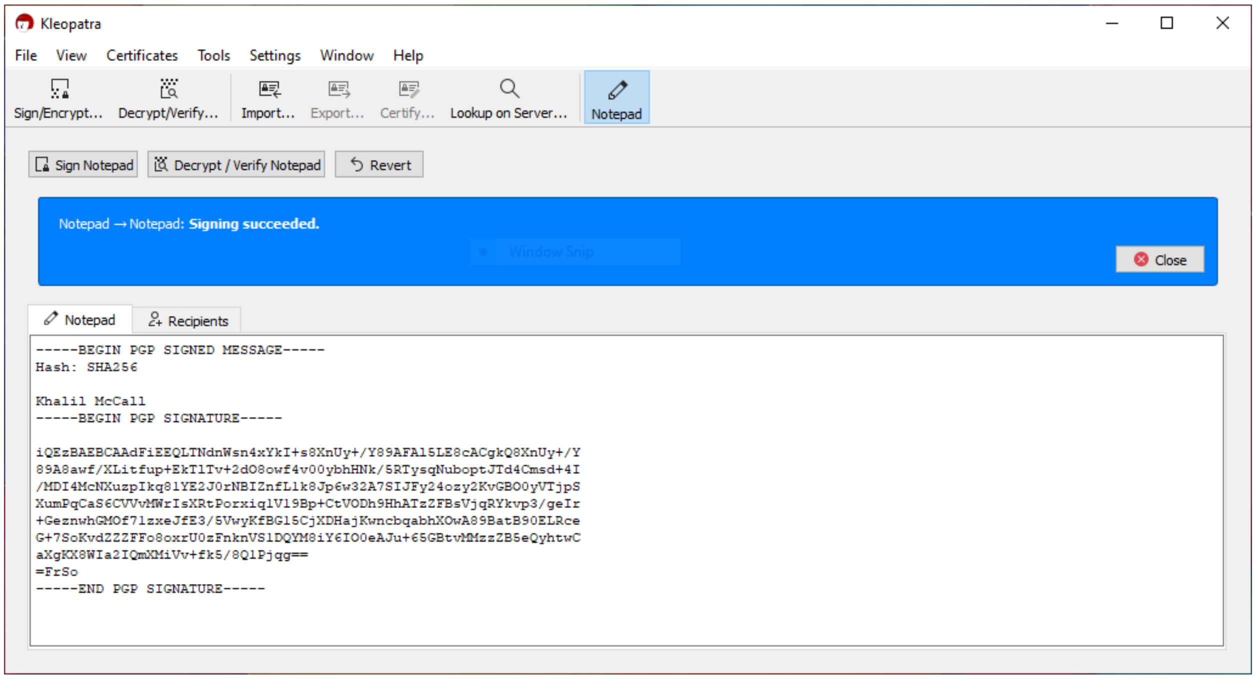


All operations completed.

Verified 'npp.7.8.4.Installer.x64.exe' with 'npp.7.8.4.Installer.x64.exe.sig':
Valid signature by don.h@free.fr

Signature created on Wednesday, January 29, 2020 7:55:38 PM
With certificate:
Notepad++ <don.h@free.fr> (6C42 9F1D 8D84 F46E)
The signature is valid and the certificate's validity is fully trusted.

5. [2]
a) Sign “your name” in text format with a GPG application.


Kleopatra
File View Certificates Tools Settings Window Help
Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
Sign Notepad Decrypt / Verify Notepad Revert
Close
Notepad Recipients

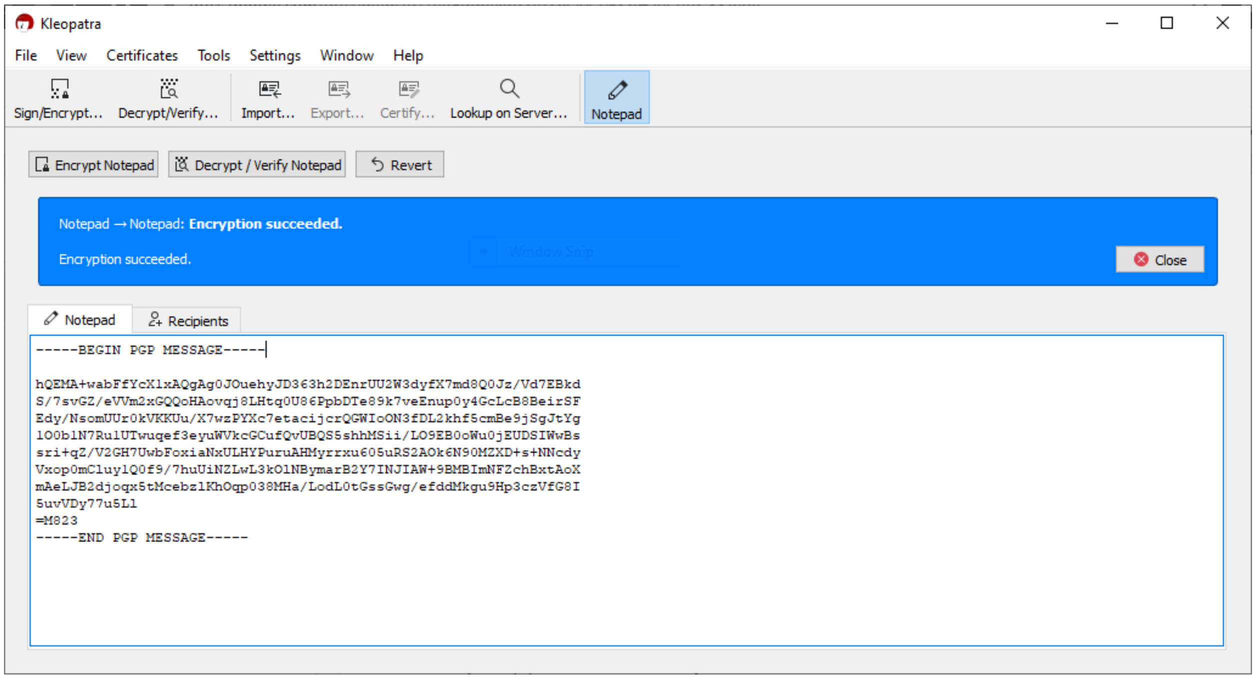
Notepad → Notepad: **Signing succeeded.**

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA256

Khalil McCall
-----BEGIN PGP SIGNATURE-----

iQEzBAEBCAAdFiEEQLTNdnWsn4xYkI+s8XnUy+/Y89AFAl5LE8cACgkQ8XnUy+/Y
89A8awf/XLitfup+EkTlTv+2dO8owf4v00ybhHNk/5RTysqNuboptJTd4Cmsd+4I
/MDI4McNXuzpIkq8lYE2J0rNBIZnfLlk8Jp6w32A7SIJFy24ozy2KvGBO0yVTjpS
XumPqCaS6CVVvMWrIsXRtPorxiqlV19Bp+CtVODh9HhATzZFBsVjqRYkvp3/geIr
+GeznwhGMOf7lzxeJfE3/5VwyKfBGl5CjXDHajKwncbqabhXOwA89BatB90ELRce
G+7SoKvdZZZFFo8oxrU0zFnknVSlDQYM8iY6IO0eAJu+65GBtvMMzzZB5eQyhtwC
aXgKX8WIa2IQmXMiVv+fk5/8QlPjqg==
=FrSo
-----END PGP SIGNATURE-----
```

b) Encrypt “your name” using this public key:


Kleopatra
File View Certificates Tools Settings Window Help
Sign/Encrypt... Decrypt/Verify... Import... Export... Certify... Lookup on Server... Notepad
Encrypt Notepad Decrypt / Verify Notepad Revert
Close
Notepad Recipients

Notepad → Notepad: **Encryption succeeded.**

Encryption succeeded.

```
-----BEGIN PGP MESSAGE-----

hQEMA+wabFfYcXlxAQgAg0JOuehyJD363h2DEnrUU2W3dyfX7md8Q0Jz/Vd7EBkd
S/7svGZ/eVVm2xGQQoHAovqj8LHtq0U86PpbDTe89k7veEnup0y4GcLcB8BeirSF
Edy/NsomUUr0kVKKUu/X7wzPYXc7etacijcrQGWIoON3fDL2khf5cmBe9jSgJtYg
lO0blN7RulUTwuqef3eyuWVkcGCufQvUBQS5shhMSii/LO9EB0oWu0jEUDSIWwBs
sri+qZ/V2GH7UwbFoxiaNxULHYPuruAHMyrrxu605uRS2AOk6N90MZXD+s+NNcdy
Vxop0mCluylQ0f9/7huUiNZLwL3kOlNBymarB2Y7INJIAW+9BMBImNFZchBxtAoX
mAeLJB2djoqx5tMcebzlKhOqp038MHa/LodL0tGssGwg/efddMkgu9Hp3czVfG8I
5uvVDy77u5Ll
=M823
-----END PGP MESSAGE-----
```

-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF4wrSIBCAC8Kjk9aH26T1/u7+NLSHNj9IAPS1yn9NI38a9hhD6Hci/V/6Rx
ytaymkZQbiaya+r3Ydh/QwaJIKUzX9Dqi5w46vTmfMScGuiyhKQQUjE3j6/aF77I
5qrOCqwRdYdEdQNiuVeyXsOJ4I+V5ZL1IaB0Dx29A5d+fY8Ukx2FiEyPj6g6EbtA

```
Ch24BE4Kb/GYPhqIXaeK9iYoIvsa2tOimAoqOGpo+U0/DYqlhbwMjuh2WZCA3c7v
/PHIal8AaQVOS7HEQGN7aVh4K+avCDPJKF78ywjivTyQ6FbQfKIP1xXtPNchpaEH
RL1WydC24TIfH6ZTLMbDkGmVEOdfS2sKb10pABEBAAG0JEltYW4gVmFraWxpbmlh
IDxpLnZha2lsaW5pYUB1bmYuZWR1PokBVAQTAQgAPhYhBHdiltBIRr09H7pMmNPU
5nva8uiMBQJeMK0iAhsDBQkDw3PuBQsJCAcCBhUKCQgLAgQWAgMBAh4BAheAAA
oJ
ENPU5nva8uiMnL4H/1QUcFq059oq1efFivjaAneI+3/P90urUbCi9lOv3m7RyIZK
Jj17Eq8peN+SVnhcW9bJ4ycB7rcDcKH8mz2KTdZIiGUtfIuv4oORv+vzwUZAUano
GYJ5tpnSzCeCCSKq+tpur8n1jPJOpQYkYE82ri/fzqaeSU80X+CbqzanZX8E8ldY
wcct5fZOIO+xEioxoWY+A9j7y22+b81UhHCEEIxIexwyv+ksIJ67Feu5IBZsJuz/
kFvVaKwsbuBwF54nw7Ee0a9JW6M4kElZk66lVCPP1SsBwgOyyodAAK2df11rE/9s
z1zdq71snGGvaa6um5EgBPqr/4A03YIylsN9QYi5AQ0EXjCtIgEIAM8k6gtQeMzw
d6f6aSU7l0I/M7xzPxZ3FCEPSaE9rq0WOKcs2Igw56WtNJqHGdkvgERv4SSKSYCi
WKdz7S7LMrK1PyU+uKujwn1yjSXFedW6J2JtqEx5rRvgpyXL1DQ4KgFQAA9wUuaU
6EsiibLAbqG0J1wZ6eQTtfeBVg8qyEHP8S13wkTN9vSSlF84fBvyRkimKsoKcouW
UCaD1n2HpdrHEKmkgwZBVwSKXLOHEG4N0L2CSwqpzofxN+hlFclJKZUYdXk41SG9
Jk8+29tQduYWyDzsOHkNNT+7YmyU5P30wYAYeKDuwqPg73fm7j2wuekYhUxaWu1T
G5HppBHNhFsAEQEAAYkBPAQYAQgAJhYhBHdiltBIRr09H7pMmNPU5nva8uiMBQJe
MK0iAhsMBQkDw3PuAAoJENPU5nva8uiM4VkIAJe9J2Mdsi9rNOHF5nvSwjsIJ40c
nNPj8SADXCAUTOUfe6gRe1DfNR+gEo/1uDRfn7hFrBFqYZl6JcDwyD2j8r6QJ0Tl
IcH3AIfdjz1PByMzVPzr8fSDV9/Nu7shEukuttSUf4sndKwNW8Sna5M1cEjvXsWd
eQTn41oETTjWpzMWcOiTBJkvDkArk4rH1eaMa7ckIvYQBuisOfhvLKiJJj9Xc/5J
69eIEyNmfFhnPBINWWxOx+us8U30vwBIrfF2UKo2YKFOie2rRGPP/NWen/ia1fHn
aNDB/nxMr9MPUgibfMFqeqHl3RuvAdf6+hxwZ8wbDAA6cjxccS1Q6ZcUcJY=
=8D/Q
-----END PGP PUBLIC KEY BLOCK-----
```

Deliverable(Text)