PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACION

Criptografía y Seguridad Computacional - IIC3253
Rúbrica Tarea 2

Preguntas

1. En esta pregunta usted va a implementar y demostrar la corrección de un esquema criptográfico que utiliza claves más simples que las de RSA, y que además es aleatorizado, produciendo con una alta probabilidad cifrados distintos si un mensaje es encriptado más de una vez.

   Para definir este esquema, necesitamos introducir un poco de notación. Dado un número natural n, sea $\#\text{Bit}(n)$ el número de bits en la representación binaria de n. Además, dados dos números naturales n, m tales que $m > 0$, sea

   $$\text{Div}(n, m) \quad = \quad \left\lfloor \frac{n-1}{m} \right\rfloor.$$

   Entonces, la clave pública P_A y la clave secreta S_A de un usuario A son generadas de la siguiente forma.

   (a) Genere dos números primos distintos P y Q tales que $P \geq 3$, $Q \geq 3$ y $\#\text{Bit}(P) = \#\text{Bit}(Q)$. Sea $N = P \cdot Q$ y $\phi(N) = (P-1) \cdot (Q-1)$.

   (b) Defina $P_A = N$ y $S_A = \phi(N)$.

   La función de cifrado Enc_{P_A} es definida de la siguiente forma. Dado un mensaje $m \in \{0, \ldots, N-1\}$, se genera al azar un número $r \in \{1, \ldots, N-1\}$ tal que $MCD(r, N) = 1$, y se construye

   $$Enc_{P_A}(m) \quad = \quad ((N+1)^m \cdot r^N) \mod N^2$$

   La función de descifrado Dec_{S_A} es definida de la siguiente forma. Sea $B \in \{0, \ldots, N-1\}$ el inverso de $\phi(N)$ en módulo N, vale decir, B satisface la condición

   $$\phi(N) \cdot B \equiv 1 \mod N$$

   Entonces dado un texto cifrado $c \in \{0, \ldots, N^2-1\}$, se define

   $$Dec_{S_A}(c) \quad = \quad \left[\text{Div}(c^{\phi(N)} \mod N^2,\, N) \cdot B\right] \mod N$$

   Responda las siguientes preguntas, en las cuales va a implementar el esquema criptográfico y va a demostrar que es correcto.

(a) Implemente el esquema criptográfico definido en esta pregunta completando el Jupyter notebook `pregunta1_a.ipynb`. Para que su pregunta sea considerada correcta, su notebook deberá correr de principio a fin habiendo completado los métodos marcadas con `##### POR COMPLETAR`. Las entradas y salidas de estos métodos no pueden ser modificadas, pero sí puede agregar métodos adicionales si los considera necesarios. Se evaluará con un programa externo la implementación de sus clases `Receiver` y `Sender`.

(b) Demuestre que $MCD(N, \phi(N)) = 1$. Nótese que de esto se deduce la existencia del número B, que es el inverso de $\phi(N)$ en módulo N.

(c) Dado $m \in \{0, \ldots, N-1\}$, demuestre que:

$$Dec_{S_A}(Enc_{P_A}(m)) \quad = \quad m$$

Corrección. Está pregunta será evaluada de la siguiente forma.

(a) Se realizaran cuatros tests, cada uno de los cuales incluye una llamada a la clase `Receiver` para crear una clave pública con un cierto número de bits, una llamada a la clase `Sender` para establecer como clave pública la generada por `Receiver`, y dos llamadas para cifrar un mismo mensaje y descifrarlo. Este tipo de tests es el mismo que se utilizó en el Jupyter notebook `pregunta1_a.ipynb` que usted debió completar. Por cada test que sea realizado correctamente, usted recibirá 0.5 puntos.

(b) La asignación de puntaje en esta pregunta es la siguiente.

- [1 punto] Tiene una idea correcta sobre cómo demostrar que $MCD(N, \phi(N)) = 1$, pero la formalización de la demostración está incompleta o incorrecta.
- [2 puntos] Tiene una idea correcta sobre cómo demostrar que $MCD(N, \phi(N)) = 1$, y esta idea está correctamente formalizada.

(c) La asignación de puntaje en esta pregunta es la siguiente.

- [1 punto] Tiene una idea correcta sobre cómo demostrar que $Dec_{S_A}(Enc_{P_A}(m)) = m$, pero la formalización de la demostración está incompleta o incorrecta.
- [2 puntos] Tiene una idea correcta sobre cómo demostrar que $Dec_{S_A}(Enc_{P_A}(m)) = m$, y esta idea está correctamente formalizada.

2. En esta pregunta deberá escribir un programa que verifique la autenticidad de un Json Web Token (JWT) en sus variantes HS256 y RS256. Para esto, deberá escribir un Jupyter notebook siguiendo las instrucciones explicadas arriba que al menos defina la siguiente función:

```python
def validate_jwt(jwt: str, key: str) -> bool:
    """
    Arguments:
      jwt: a well-formed Json Web Token
      key: the key to verify the validity of the jwt
    Returns:
      valid: is the jwt is valid w.r.t the provided key?
    """
```

Su notebook podrá definir funciones auxiliares que ayuden a simplificar la lectura. Deberá estar explicado y ser fácil de seguir para una persona que entiende los contenidos del curso.

Importante: Su notebook sólo podrá importar las siguientes librerías externas

```python
from hashlib import sha256
from base64 import urlsafe_b64decode, urlsafe_b64encode
```

Para validar un JWT que usa HS256, el parámetro `key` es un string que representa la llave a usar en HMAC-SHA256. Por ejemplo el siguiente JWT es válido con la llave `IIC3253`.

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjbGFzcyI6IkNyaXB0b2dy
YWbDrWEgeSBTZWd1cmlkYWQgQ29tcHV0YWNpb25hbCIsInVuaXZlcnNpdHki0
iJQVUMgQ2hpbGUifQ.Cn1AACqINaUTbAJuh_V4lBcr9X4dRp8FUX9sGDkX-Ss
```

Para un JWT que usa RS256, el parámetro `key` es una llave pública RSA en formato `PKCS#1` para validar la firma RSA contenida en el JWT. Por ejemplo el JWT

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoiQ3JpcHRvZ3JhZsOt
YSB5IFNlZ3VyaWRhZCBDb21wdXRhY2lvbmFsIiwiY291cnNlIjoiSUlDMzI1MyIsI
nVuaXZlcnNpdHkiOiJQVUMgQ2hpbGUifQ.CmoMizX-_E2Ugd7-JDThCfrXTJbg38W
Lal3HipmnA8oAUh1yG9IU1n_klJkmPIT3knxrmrJMXxh6gTCOylLQfKSQI7pHsYUr
-yOd5gL7XpnT3stvOtYDO383cBnrL5X8EV01lUxJJenYG5Qr4uVG7Msg-4fUJbTqT
R2t0Jx2UQ2pfi_jxgfg6lAjSLK9TygntJJ-eJV0Q8IipVYnqtCxBS-OIXekalyjpB
Hksf_ibiJtPrMJI3Kvyj3dwrETth8c4yg2IIh22uoJHrJArNk3xPfeSsasZT0ixfM
E8Mlnkd4HwpbcNZZl-FpsBPbPWHKynZXptq8uS65PxKmTggg8kA
```

es válido con la llave pública

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAu1SU1LfVLPHCozMxH2Mo
4lgOEePzNm0tRgeLezV6ffAt0gunVTLw7onLRnrq0/IzW7yWR7QkrmBL7jTKEn5u
+qKhbwKfBstIs+bMY2Zkp18gnTxKLxoS2tFczGkPLPgizskuemMghRniWaoLcyeh
kd3qqGElvW/VDL5AaWTg0nLVkjRo9z+40RQzuVaE8AkAFmxZzow3x+VJYKdjykkJ
0iT9wCS0DRTXu269V264Vf/3jvredZiKRkgwlL9xNAwxXFg0x/XFw005UWVRIkdg
cKWTjpBP2dPwVZ4WWC+9aGVd+Gyn1o0CLelf4rEjGoXbAAEgAqeGUxrcIlbjXfbc
mwIDAQAB
-----END PUBLIC KEY-----
```

Corrección. Para evaluar esta pregunta se utilizan los 20 pares JWT/llave que se muestran en el Anexo A. Se evaluarán 40 pares: los 20 que aparecen en el anexo, que son válidos, y 20 pares inválidos que se obtendrán aleatoriamente sacando tokens y llaves del mismo anexo.

El puntaje obtenido se calcula como $6 \cdot r \cdot (0.7 + 0.3 \cdot d)$, donde r (ratio) es la proporción de JWTs correctamente validados/rechazados y d (desarrollo) se calcula de acuerdo a los puntajes que se muestran más abajo. Por ejemplo, si de los 40 pares llave/JWT a evaluar se consigue validar correctamente 30 y d es 0.5, entonces el puntaje obtenido en esta pregunta será $6 \cdot 30/40 \cdot (0.7 + 0.3 \cdot 0.5) = 3.825$.

Cálculo del valor d:

- [0] Entrega un notebook vació o que no aporta información concreta respecto de cómo se valida un JWT en las modalidades RS256 y HS256.
- [0.25] El notebook explica parcialmente cómo programar lo necesario para validar JWTs en una de las modalidades y no explica prácticamente nada de la otra.
- [0.5] El notebook explica de forma concreta y correcta cómo programar lo necesario para validar JWTs en una de las modalidades y no explica prácticamente nada de la otra.
- [0.75] El notebook explica de forma concreta y correcta cómo programar lo necesario para validar JWTs en una de las modalidades, y parcialmente cómo hacerlo para validar JWTs en la otra.
- [1] El notebook describe de forma concreta y correcta lo que se programó, que es exactamente lo que se necesita para validar JWTs en las modalidades RS256 y HS256.

A JWTs y llaves para evaluar la pregunta 2

RS256

Ejemplo 1


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6MX0.P2NRWFUwAMQjQeBOuEmEv-
F8OELEQBFWU2-
lP8cSXcdZJFZClMjn5SjuqhwOUhd8InBz1wBHn6fC0ZGP0b6cP5UCZ7f2WNq3i5Z
8lTBmI7rP2NRWERHiSC8nMr1iAidBDFyn8kT77nnLVRKv-ypROFDzc-
q5YtQVYKsk_1TMnza_p8K7r42A5mKQi2fSqSPmEjR1mG3mwxgqUHHMfN6YPxorup
9L15pzo3QNdOFmgp-kakLpOlMQndp_DaVvx-
w74RQQrL8fMcOGiZc1b_Q99INQIfow3YDUM_lBqP_sF6qSK5Ws5kbJQFvXw7tLeF
RbGU4W9sut4Qt1PTj4Jo4Dlg
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAtPG0rHLP1AOHqfZZ4RkM
Fr6cUo2KgNdzR6ifQ4oddHveoArMhpVT4hA7hf/N8k+WklNCmsJWGGTnS8Jd6Wbs
Vtymszh9ZGnb5ei0+1tz5qhrp5Cq/qO51YQ5CpqhA626rbQ9kLBUbJ99lQXPBcxq
Acj7OB4r0pkA4qRqyOkddU6Czm+K5TvH6luX1ZlxUXAHiOO0KqoQ9KEhbqaphGJc
Y2YPcGWkv2y2E7DwS1/7QnxA4ge3uSyLg5/ZsmFQ+BFp16kYrN4jz2krGJYqMI3V
LuF6g6SxfnFA5RbqlG9nlKTxesowgEvh8RPFitSQCBgSqsI3GoMVEetqlGXIfMko
iQIDAQAB
-----END PUBLIC KEY-----
```


Ejemplo 2


```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6Mn0.HPmVOvBojOFshW_oJnnFndF
OKb2UBhDuPPxD8xdd7N8avaAOBrlz6P3W_WTZQ86bRtg5ZOZ0fX_KZVsB-
l0ub0_o21Op1fyMw0-MsEIsxEzKmG2oHjxLQAoPsZwNsnz-UZeWg0HDkMzOKU_pB
PN0mjgsRj9uye7QfPMqSxcp7pBvYrxAYoHjBEMXvDiGCxFDOu6j99LagmUyHwjY4
uLYsmktIP4mhMVHcaCLIV8WWCzQNuvT2hztkEoidjPExYCIb3Py4gKr20V22EalV
H0I5aFA5Loq9tR7L3iPEO438bnolfPgRxNVqfwYwSicVdLVqhz3zH_isRgh3oZKr
6C0LA
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEArPjW3qHiQwdbklYNo4pR
v/CoVkRfVKe39YgO9pi8BihArLg4tqxExbL9LMvJRJn3EK9KMS0cWygnwXA0A59I
hAa+P+PbLK4dgsbU4z1c2ubma3mBkFs6R/9sgMjOHTfpAb+QdVQRZz7D+UNvZ5Jt
WV0EhODR+GMe2srH7rAKDRWXri1AOJsRo3TJ6xG/lSM3S+u3AA5Fsf6Zr8gTy39r
dfm0yCmteSdv2+rPzdu+xEYeiG+U/54Jfu4TValpsD0PX4Yu+6//uwSNgRQPb9r9
s4ZhNrH8KKbQZNNQtKWCuvkUeD1xvxldWDoL+cN2owsA2G+pKosC+tgjy1P8rKK8
twIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 3

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6M30.nP337N4siZevBACjIee0G5D
HvxowuNcmbVSK26UvvlylU2srvIQri2qDMq07axAr5FADAFf8ozH_dvPk-
3fWRJgj-DQUqUTTkxNuFLAFAoRErhDwuzpTlJ8rsr3TFCO6SDqo4ASpnDE-tjVnX
h7p50EDEsuRLr7GNYDX_29rAh_uCjJf1NNM7nWenHf_3jxDqAWwHscBxiZQnOLoK
q6hs29oSG5-4GFDx3g8hsg20PNfjdQNDMhijZ4WCsKiNr5sQ1SiB0zloGcnMXy-
8HJ5JaYqJGmNJabm1NYB_aghGnpgNonEKnObdIDBa_FImBednqmthMUMjnUFoInb
Q9pnLA
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAq90Auc37onA8d+/cAcCv
xdlqE1wpZNYXF3u1invZhHRrXVaKJMSrDS8FotHG4VpCNKB+sV8Bjrg/2R1IJB/k
dg1ffzx7T5HPzseamecl2LRpUgVsnQF1P0VnNaxdlK0avklYwApZ1wetZoZUUmYk
CF0zvamUMgguaIcs397LUAi7fwug2GO4teHZSpJa019Hi7jDol9dKZviMNEL//bx
pma71M+a6wbnaxITsdgukdVXueh04qffqc46OSy9yjn9+6Gbacz0IEccVBlkxkcg
7TopI63FFXW26nSPBKrC7bVTrp0Q+Bn61x5XNAyO6YR+h9FXrjPyi84KjWd3yM6f
IQIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 4

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6NH0.Md6O7lQAmr5hz1xe8GVRxck
nB-bxXVDXPvn9xLjJvAA09hWgXuqCYIO6JBz9SsvdaHQIXtPO3Ptcnn6Bqbf66iz
FZD413XQnDsMZWNKW4G-
4xBAy6y2Oo28rQ6CLGbLJ3y7r4A4Rgl1vOuUxMVml8XOx_V1f12MmGSM6Mj8D5t5
CZnVebNMnBAIw0BzsmxA5Th_OSnts6kkTXztlaCP3w1JhX0_v0c5NXUzDKj3qZru
9M62mqXhlf3iAsT9CIMO8AKD4CZabCZ40IkTH3mjmvvLthtDtVQQ545tpzSDOl4g
9kix1H2CylSm3LkZUxTy9_UWYg6KWkTcRONl0YKagzw
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAstPTZU8HHR4dHYhCAAkK
dvXAtdQG71z8UlrjWt3BvYzOPZ42jdebzbUdgwF8yd47nzwk2zfDI4ALEpnzQeXf
wXNKuiF/lEgeZ3e6zRgkDbNDMbXPYWcncnvgW7rld2kVnubPB0I7VdP1J/fUJvUN
0vLo7MeC7abbpuDaktkXPowDnz3X3HwViYhrw/xt7F7P4BxTg3DGduqcIV+tYqLK
6HWe4RFvRjgXj54v6XQxi1i2BGikDdXoYm020EsYcQ7FPDAwAqC4ppKMUVZnJZtk
+cdxHgF//ZIkGcAOd4+UTVVV4SdNPxmK0rC9yKVPDGXKI3x5LNPoUnaMzM9ueXoQ
iwIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 5

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6NX0.VVML1ON5dx0g8aQ5ZIANM9U
6KCmt0FPvhFz-XFCOgKTyfbA0mwArFg6m1ODPtXLP_xtak_GeNF0C8fgv22KizZU
```

```
4E_7znqAHKgsZs8KO7tgq8YXnKIMIjfK--VKl8jfTaedIKVuW3aOn2XiLuPO9DNU
7KgLZz1KRaBRAc18Pvdjx_ccvurqNNapY7RhVZhsnol-
DPge4l9ZIaQ1Ur7NP_DGDAJ9n66xKGek-PDC-_0fx3IV8JSMOLwxmnaNcQPMvz-k
vqWwxWA4CkMb2PXwUn2iyC1Ho7k1-oWxVSKdcNsqi1sP-
k1wYd38Z3e3ub40omK9C7oWXzaHE72FJCRgfOQ
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA4HA6t47oK9iEC2QQZRcQ
+m98jBDA7khz8YmAI3GLiHsHJNd6x44frooXPOVxaK+HdR3XXNWjRK+fWd7oa39A
Tv4Ys+upIXRDf4P9eGHybP6FgWaWDqDDqSN/wbqP/tW9O1MHHCjvKC8cmLMpV0vh
f/AviWPABhGIn4xSdelBNUeu9YLIOtc5v9++gWN7fw9FgcBvP81GiLlRxHYDkChh
g+6CHtGytLklBz/010NDz0EJX3BX5JYibIeb2ScOZu2lAqaSQrSDY90UH4KmRP6h
BgAHVTvoAhRnmKvn4QIzrVdjI+AWi6PxywRo0+mo5G4v46UvdBig5RKZjBzJXd6z
xQIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 6

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6Nn0.moVogFivAWiKYc4u-
o0P_Lo9Y7ePibJGMaWVPeRtJ34gvTpbDRwH58TWeqbHfLRn6aLjaCQfmzF-
WPfU7Dyk1OwxVPtQwfiv7a4e8xDHKhli-yJAHt040rvV6RccEYZ2D-QBW-
7Mdw3YyJ0Yf5svDYMLIy-
1j7r3l5Z3_SRBmpRyrjayR3cLIRy3InW18RaWw4AjZhGjJ7gzD43QuZ9k-
nUtoPxkYhQ3YzGea3eM1yilEaYO5-wZTLC2rD_lUHSuzlpv4-
h75JUOSHYQX465XNaihF2X4HOVHD41-8WWbXZvSG2ckP-_Bq5cK9t1q0jslPqUj-
zKHUfbOGJ9PGdh9w
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvA4LpJnCK5pJM8A82j0V
g8TeZ4ySVhHVIn4icsPPZXAnBjBIMlGoGJCqcec5ixUBtbRIAOZDeHqxQG8U2A+d
9m9ujmch/2B1GMeYaLHa88JrmoV5+E7T3u7b+5upt9N7oEJDsFIn1nMh5PGIJNBq
QFL4h12IJJO1p418fo/VLayxjsPz+WigkvJe09jkl4GYDQuYZ8nvtJEgqOfv0WhH
Rw+yFAxfyNUYIB7eXDJJxncv9i9uH9E8SNQ4tEKnIVsukYpeweUeoDXdHomykCaU
w0x3tKKrkkJBJcJkbTpY9qxpIC1y141ls6OR3guSgGBLM8bTnv6f5VjrABdUYEYg
JQIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 7

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6N30.arE9GQetbkXv4OmNtt069Wt
ghum3vvhGTkLoBAMjJ1EDy2uQBCFmPDKvIMUx7qWdcFerSo9U0JuvgploAzRF4Sm
IhnzXid28LuDeGY4JRXTTgCmHqaWiwqgFJ2f59C1NR8docnPuXfPsB67sgB_tcIH
E1Bdqf2LRrRctr6UUTIbjJkJOoibb9I1Rc7OkEQsFfkQcPVJTgUPtb8bqIuGuQIx
bcG68h3mlqsdkc-YFcNJEQd8z5hhbUAfPDRSzcxHCA2iNYLIGNB16dLcNi_qpIor
9Anlowaz8bOJosNdh_BgObIFaFMrT69Uy6cxgPIGp0-6Xz7XbcKU9-PCO8BxfgQ
```

```
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAimtr/lhjCUajvn/sEbiI
LS4Wr4q6SwfZKWJx4+xwkwB2/bgyXYX6hVISAW63M5245+ipGp4FfnawiNBNafRM
fh37Hm2UK0/UpGxVSUp7Npifufn+wchqip5lloP9XoroCOkUFjM1+SRebubSvPoJ
2SSg+nXthfgguC7AQE/kWSTIzbx7A/m9vkIRVJGPVSCOkUhcXtJzXeBO5oziOlSD
rg+HO5AREdT2B25ak2ygT8s7S/MzMzY5dYbftwQ88or8Bq/ltic6X8xl7x9aJw5V
fNj0tbRGH3cbfHllwDIDySnFPbpeYHQ8RMsXk3TT2/ULadQT8FydNFnwqcLxkRcH
cwIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 8

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6OH0.BXxH32K5YL5E1-
DykCub7AOAgM_QoCro2IY7AqjrkH9dhTu8TK5SIgVBQy_7q_Cl2I49z6xX9Xdf4L
j1qS9ZY6y6JHSL98skLPX518vWyn7t--
BT12CxeWV2RcSJdKUvVY279beBZZBSRUDa8IV-TAu9H1-zmRYuC06Jrr6L-
gaRLPlPTsl9A-Kzp7-HQS2k2vFw2IC9Tzy7k0UQJTTPW-g7tl4-
q3Kgb9e34jiqL_VwJQd_3MxvSwEb_wmO8Rm5jjKFZsx3YJkU1_ysd-cOYzbT5G7m
53fW0CRQNe2Gr2B7GPeSiHtPE6Zh_yPv7J_Mq9msvbLlytLQiO1IRrpBEw

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAmNz4RM5vA/eli86SAskF
vtKHdQwkwTLE6LLBJqxID9I0JI8uZNM6LrsH1GNx4kfwRXawajv6L2sFUq01pxT+
5NSIU/HntFbOk8BBLNGtxfAdR7c9Juf3BUJqvTJCphssJmSzCcPsXDK5Uk9TcKST
/j4PXf7Hzh1KH/KvTrtTEWB6rRR/L/DZ7ItwooF5mYVI2rH9TmShP5v+XHB1UZm2
pSOuZKFBySdPzXP23SKJmMl2JQvU65xncpirRK2aG3C/Z0vLx3ee8sVSLhQGcVGH
lmkF91s2vUxro3fSc4fLZhIUOXfi2SmwSDZpsno2OvAmPnxqGeUKXa5hM1rnIggS
uQIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 9

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6OX0.Zyl_e-mhIP0VaoXS5REcaLZ
tZXTaBXv3btGIL0LLbxTd7xVJjewzu1Fh4U8ltpk8-qmmO-
N8GI178I0QmMFvwugF1VGSjypTQkerNLHI7xQ15tL-rNxLA3DRJnkQzpXfp7zlrO
aoEypaRXQlvHgQvm89to881U6_rIohr6x0sblYWV3X3WyQRA0EMBztZk3gsrnSv1
OGSo1u8hlLrfySxDxB0rl2YDObLMwuk-
ZKleYudX88CRVDwZkrZahMU3fuP5pBvnpcB9A7eQ-
sUDgrEAHFookzPfLZTfp26aXUXCpllhkelBFoqaCJgxxGDfFM-
_sUg565UOs2UGYGmmZ1MA

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAr03ULGQ230BgrbnVzBbn
AlvUvVkNn0g/GsE85WKYizvCGH7WjuSfO0Efp8IQfjI8lkh7V9vv3Wue3smbB+1W
```

```
KAlzHfWVzTdjasrWvg/YuTYxwuO3Ep6f1m0tqkzwBzC0HbuaLVl6p3RLDcgTvdwx
voh+tG7R0DjZ5GfvmZzB37tPFefcZ+SQskY6s+/4Lucyr//RbZQQWKNmsFnmialB
98cENx9qWnrEOPgwrF6Opsh+IKM1aIUuytXMDolwSr382go9fjA571oOb1QdYNSa
iaYDakWjg1oAD70PSwioyRB6YHI6G81hFUWiACUxEDN4bc6vK0nMzWCSp8z0lM39
gQIDAQAB
-----END PUBLIC KEY-----
```

Ejemplo 10

```
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjE3Mzg5MzYwMSwiZXhhbXBsZSI6MTB9.Oo4ep-AnwNhAa9u-
6NHNubB6AIHTZtPsrrpYQ0R1ccTK4mGBiSnr99WuJbKHxAbhRXyyRWq50d-
mFgMk24wDAwbqwehWoyH19xmoZY3_WXuC-
26QJI6M1bhfOcRRbkYu_WhkEKVsv3M_yV6m6yDTpBkb3dxgrv2xgk-
Q9h9RlFybPp2F3GdHRvKG_HmWnZdhsaIOUkO_-
bibpl_LdXZE0LoWVrKIBMgPevsXqMkfCtACHJwxKaw-rKq5ZpuULQLaNS50eFAyz
k6DiAo32viXO1K795cNuW0ObJ8ASUJrvaHjP5QHKg9B-
Eu_XuCS_s9fK6z5OUQC5233wgWByTAPmg

-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAoBrpXebvHb5EGXsZjqtz
s4tez3PnIX7M915Yq0vs/7fMheFpp36JXf+oArufHlWWW/BrAv6JeMD9HgR/QVm1
l+9eI1382blIr6GRjGAIIlmksq8hUKOrWm5gydh+TGd6qYlR23Sha/VAOr+DWEI4
y+fi86DF17656YOKIOkVPUM8vBsk1ijUcZ8yGIYkzMSzLct1aOD7YWGDudYpvoS6
d52cV+258Mft4TxgjC937Nj33x8/aMNKCDR/99V2ubfUM8M01OQAyxdsXyDDTP4n
riT9riu76B8fHdVNIw7niBxEXjTdc6LoVkYis1ewmDmxzr1+IIrnjMiZQL0JF3gB
ywIDAQAB
-----END PUBLIC KEY-----
```

HS256

Ejemplo 1

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6MX0.C1cvfudj_lFp6Dk2qVYHtgy
IcKf5mB1H91NHdKYexjo

bfd167f8-1f6e-49bb-b152-eeaf0eda3f36
```

Ejemplo 2

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6Mn0.-
Ob9Ze3wPk_6XDYYjYrKJGOND9iOed0T8wg_TjB4DnI

502a9d1f-6fd1-485f-9901-986ccf804781
```

Ejemplo 3

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6M30.8lLpAUUSpkwhFT_HYlVekZ8
r7-tqyOcsKKGkGNpRs7k
```

```
895d1c9b-e0be-44bc-9f73-58cee944f0ad
```

Ejemplo 4

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6NH0.xQx_9KizFR1pMmIrc6PB8yC
TxcJJAI4cdnGHdYGLxpw
```

```
ea7a6079-0313-494c-9327-d7b7c43caf79
```

Ejemplo 5

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6NX0.o7Gv2tnihXIxGWYseHIoyyE
3AOZwLMSlof6QSOHONhw
```

```
7675ed90-c7ba-4f82-b6ef-e41b08ba3de7
```

Ejemplo 6

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6Nn0.e2odKiRUXK0uKB6x-
eCuDNktR4Pu9jJgNFc-SyiDKaE
```

```
610d27c5-0c4d-40b8-be54-fb60e37d4435
```

Ejemplo 7

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6N30.fl6DxsNisk7Ndptk5B1aZIU
kggck-fHOm11L1ihhxo4
```

```
efc7d850-cdc9-48a4-92ee-ef956df0e62b
```

Ejemplo 8

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6OH0.AnssILhPuKQMf1c3z3mhwTv
KvxkjOPPKsypY5rVHZLQ
```

```
2d7b8740-5bf4-4f8c-8af2-9b68833eaddc
```

Ejemplo 9

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6OX0.0pLjsDEVgxoBZBvpTD-
kF0p9CzRGjPHTUD6R8DwnO7c
```

```
10cc22de-6b9e-43bf-b1cc-875a10224d36
```

Ejemplo 10

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJjb3Vyc2UiOiJJSUMzMjUzIiw
ibm9uY2UiOjkyNzU3NTkxMiwiZXhhbXBsZSI6MTB9.dVYlJU-
7kNxiq3VVLwOxVVGnUx5BTfnkUGar4Oc6MRo
```

```
fe1fae7f-d0e8-43c2-83b3-ef21324d290f
```