In [8]:
```
pip install matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\umute\.julia\conda\3\lib\site-pack
ages (3.3.4)
Requirement already satisfied: cycler>=0.10 in c:\users\umute\.julia\conda\3\lib\site-pa
ckages (from matplotlib) (0.10.0)
Requirement already satisfied: numpy>=1.15 in c:\users\umute\.julia\conda\3\lib\site-pac
kages (from matplotlib) (1.19.2)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\umute\.julia\conda\3\lib
\site-packages (from matplotlib) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in c:\users\umut
e\.julia\conda\3\lib\site-packages (from matplotlib) (2.4.7)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\umute\.julia\conda\3\lib\si
te-packages (from matplotlib) (1.3.1)
Requirement already satisfied: pillow>=6.2.0 in c:\users\umute\.julia\conda\3\lib\site-p
ackages (from matplotlib) (8.1.2)
Requirement already satisfied: six in c:\users\umute\.julia\conda\3\lib\site-packages (f
rom cycler>=0.10->matplotlib) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

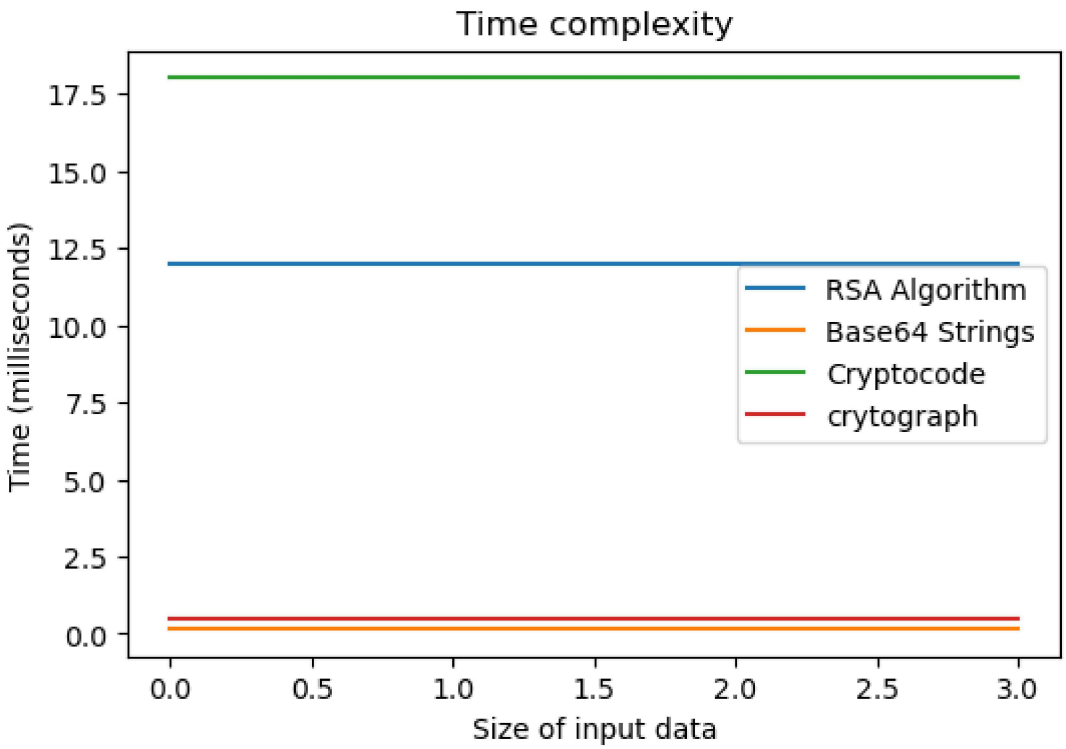
In [10]:
```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.figure import Figure
from matplotlib.backends.backend_tkagg import (FigureCanvasTkAgg,
NavigationToolbar2Tk)
```

In [11]:
```
Interval = [0,1,2,3]
#x = np.linspace(0, 100, 10000000)
rsa = [12,12,12,12]
base = [0.1607,0.1607,0.1607 ,0.1607]
crytocode = [18,18,18,18]
crytograph = [0.4755,0.4755,0.4755 ,0.4755]

fig = plt.figure(dpi=100)
plt.plot(Interval,rsa, label='RSA Algorithm')
plt.plot(Interval,base, label='Base64 Strings')
plt.plot(Interval,crytocode, label='Cryptocode')
plt.plot(Interval,crytograph, label='crytograph')


# Add labels and title
plt.title('Time complexity')
plt.xlabel('Size of input data')
plt.ylabel('Time (milliseconds)')

plt.legend()
plt.show()
```



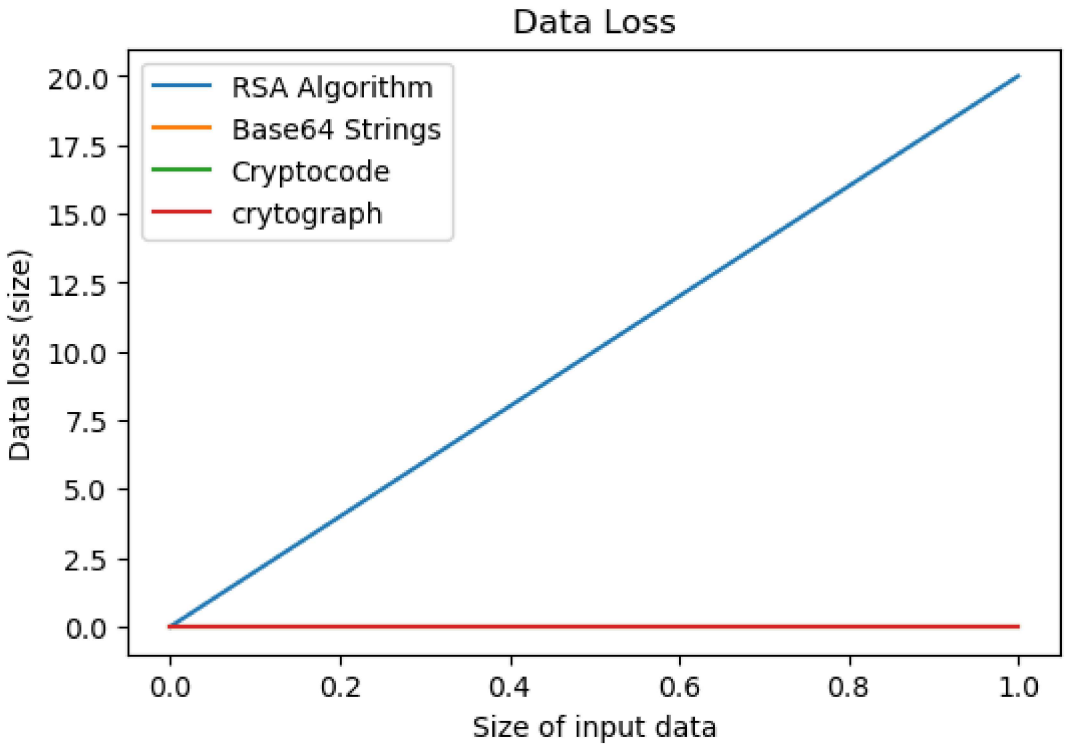
In [12]:

```
Interval = [0,1]
#x = np.linspace(0, 100, 10000000)
rsa = [0,20]
base = [0,0]
crytocode = [0,0]
crytograph = [0,0]

fig = plt.figure(dpi=100)
plt.plot(Interval,rsa, label='RSA Algorithm')
plt.plot(Interval,base, label='Base64 Strings')
plt.plot(Interval,crytocode, label='Cryptocode')
plt.plot(Interval,crytograph, label='crytograph')


# Add labels and title
plt.title('Data Loss')
plt.xlabel('Size of input data')
plt.ylabel('Data loss (size)')

plt.legend()
plt.show()
```



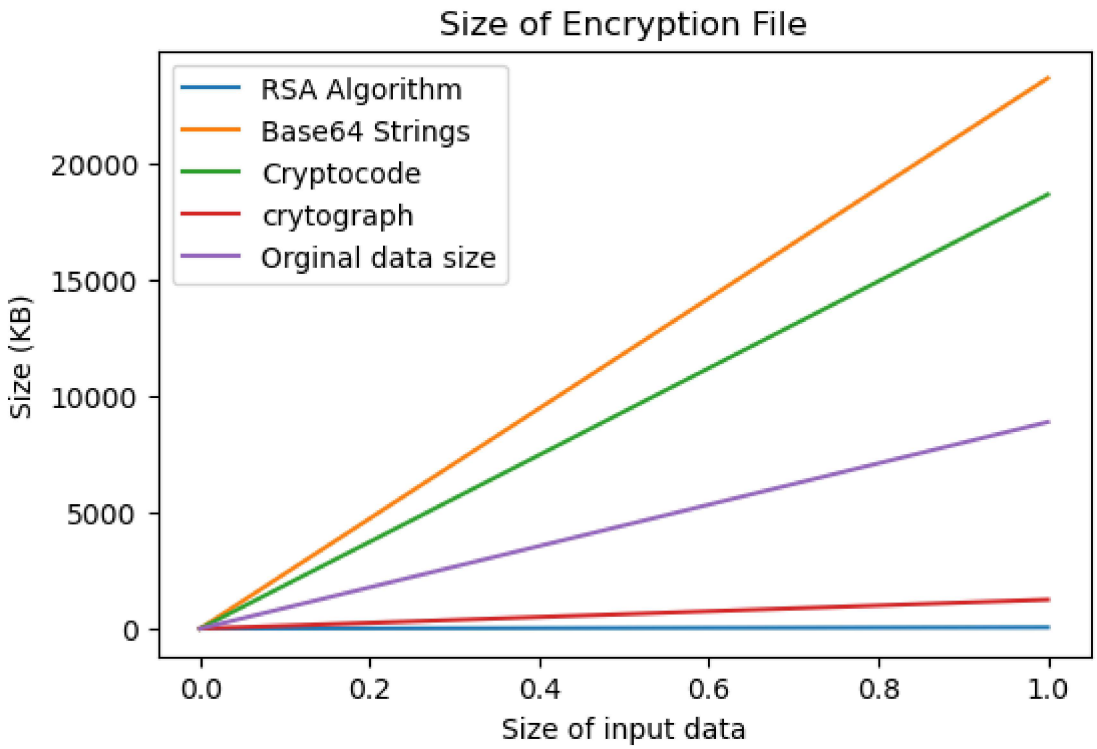
In [14]:

```
Interval = [0,1]
#x = np.linspace(0, 100, 10000000)
rsa = [0,56]
base = [0,23648]
crytocode = [0,18648]
crytograph = [0,1245]
orginal = [0,8874]

fig = plt.figure(dpi=100)
plt.plot(Interval,rsa, label='RSA Algorithm')
plt.plot(Interval,base, label='Base64 Strings')
plt.plot(Interval,crytocode, label='Cryptocode')
plt.plot(Interval,crytograph, label='crytograph')
plt.plot(Interval,orginal, label='Orginal data size')


# Add labels and title
plt.title('Size of Encryption File')
plt.xlabel('Size of input data')
plt.ylabel('Size (KB)')

plt.legend()
plt.show()
```



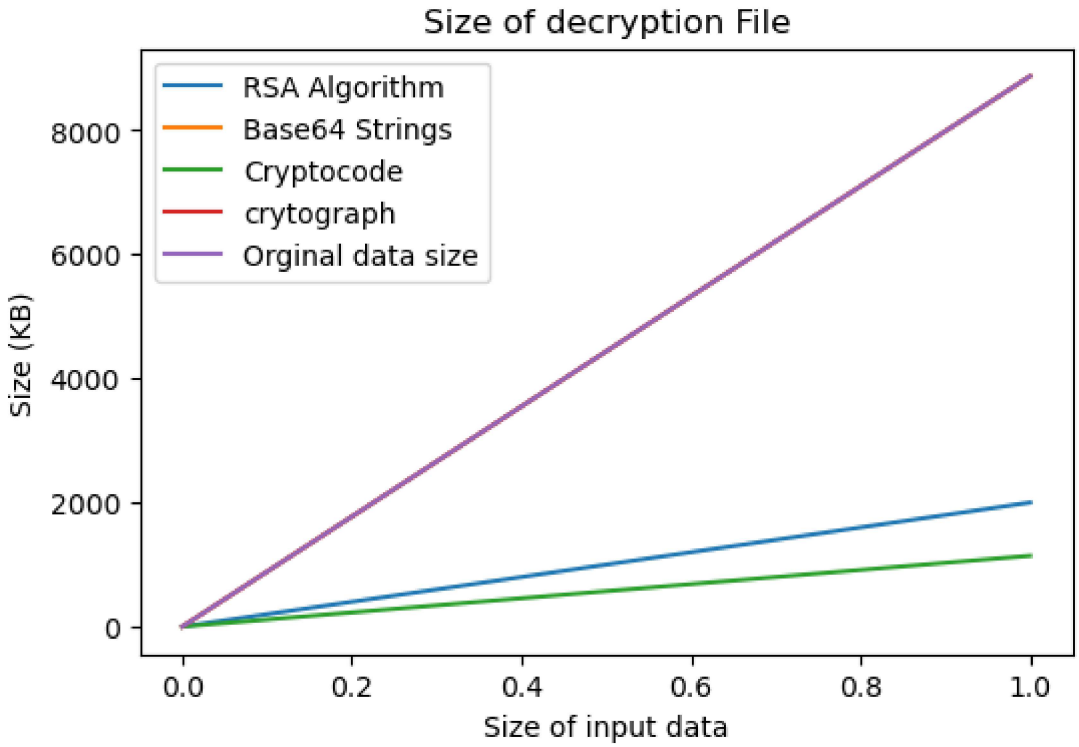
In [15]:

```
Interval = [0,1]
#x = np.linspace(0, 100, 10000000)
rsa = [0,2000]
base = [0,8874]
crytocode = [0,1143]
crytograph = [0,8874]
orginal = [0,8874]

fig = plt.figure(dpi=100)
plt.plot(Interval,rsa, label='RSA Algorithm')
plt.plot(Interval,base, label='Base64 Strings')
plt.plot(Interval,crytocode, label='Cryptocode')
plt.plot(Interval,crytograph, label='crytograph')
plt.plot(Interval,orginal, label='Orginal data size')


# Add labels and title
plt.title('Size of decryption File')
plt.xlabel('Size of input data')
plt.ylabel('Size (KB)')

plt.legend()
plt.show()
```


Size of decryption File
RSA Algorithm
Base64 Strings
Cryptocode
crytograph
Orginal data size
Size (KB)
0
2000
4000
6000
8000
0.0
0.2
0.4
0.6
0.8
1.0
Size of input data

In [ ]: