[CSCI 55500] Homework 2
David Xu

All attached code was written in Python3 (3.6.1) using built-in functions.

I. Jacobi Symbol

The algorithm to determine the Jacobi symbol was implemented as a recursive function using the four properties from the book. Each step takes a number in the form $\left(\frac{m}{n}\right)$ with an optional input s to determine the sign ± 1.

$$1.\left(\frac{136}{457}\right)=\left(\frac{2}{457}\right)\left(\frac{68}{457}\right)=\left(\frac{2}{457}\right)\left(\frac{34}{457}\right)=\left(\frac{2}{457}\right)\left(\frac{17}{457}\right)=\left(\frac{457}{17}\right)=\left(\frac{15}{17}\right)=\left(\frac{17}{15}\right)=\left(\frac{2}{15}\right)=1$$

$$\begin{aligned}2.\left(\frac{34333}{532789}\right)&=\left(\frac{532789}{34333}\right)=\left(\frac{17794}{34333}\right)=\left(\frac{2}{34333}\right)\left(\frac{8897}{34333}\right)=\left(\frac{34333}{8897}\right)=\left(\frac{7642}{8897}\right)=\left(\frac{2}{8897}\right)\left(\frac{3821}{8897}\right)\\&=\left(\frac{8897}{3821}\right)=\left(\frac{1255}{3821}\right)=\left(\frac{3821}{1255}\right)=\left(\frac{56}{1255}\right)=\left(\frac{2}{1255}\right)\left(\frac{28}{1255}\right)=\left(\frac{2}{1255}\right)\left(\frac{14}{1255}\right)\\&=\left(\frac{2}{1255}\right)\left(\frac{7}{1255}\right)=-\left(\frac{1255}{7}\right)=-\left(\frac{2}{7}\right)=-1\end{aligned}$$

$$\begin{aligned}3.\left(\frac{467827}{112233441}\right)&=\left(\frac{112233441}{467827}\right)=\left(\frac{422788}{467827}\right)=\left(\frac{2}{467827}\right)\left(\frac{211394}{467827}\right)=\left(\frac{2}{467827}\right)\left(\frac{105697}{467827}\right)\\&=\left(\frac{467827}{105697}\right)=\left(\frac{105697}{467827}\right)=\left(\frac{15619}{45039}\right)=-\left(\frac{45039}{15619}\right)=-\left(\frac{13801}{15619}\right)=-\left(\frac{15619}{13801}\right)\\&=-\left(\frac{1818}{13801}\right)=-\left(\frac{2}{13801}\right)\left(\frac{909}{13801}\right)=-\left(\frac{13801}{909}\right)=-\left(\frac{166}{909}\right)=-\left(\frac{2}{909}\right)\left(\frac{83}{909}\right)\\&=-\left(\frac{909}{83}\right)=-\left(\frac{79}{83}\right)=\left(\frac{83}{79}\right)=\left(\frac{4}{79}\right)=\left(\frac{2}{79}\right)\left(\frac{2}{79}\right)=-1\end{aligned}$$

II. DES CBC

The decryption algorithm for DES in CBC mode was implemented as a class object in Python with two inputs IV. and key. The hexadecimal IV and key were first converted to integers using a base 16 to base 10 conversion, followed by an integer to binary conversion to 64 bits. For ease of manipulation during permutation, the bits in the key were treated as string objects of '0' or '1' in a list, for example,

```
1111001000001011001100011001011100100100101001011100100000011101 =>

 ['1', '1', '1', '1', '0', '0', '1', '0', '0', '0', '0', '0', '1', '0', '1', '1', '0', '0', '1', '1', '0',
'0', '0', '1', '1', '0', '0', '1', '0', '1', '1', '1', '0', '0', '1', '0', '0', '1', '0', '0', '1', '0',
'1', '0', '0', '1', '0', '1', '1', '1', '0', '0', '1', '0', '0', '0', '0', '0', '0', '1', '1', '1', '0',
'1']
```

The key schedule was built first by initial permutating with PC-1, following by 16 rounds of key shifts on the left and right arrays of the key and permutation with PC-2 and once again treated as a list of strings:

```
K1 = ['0', '0', '0', '1', '1', '0', '0', '0', '1', '0', '1', '0', '1', '1', '0', '1', '0', '0', '0', '1',
'0', '0', '0', '1', '0', '1', '0', '0', '0', '1', '0', '1', '0', '1', '0', '0', '1', '0', '1', '0', '1',
'1', '0', '0', '1', '1', '0', '1']

K2 = ['1', '1', '0', '0', '1', '0', '1', '1', '0', '1', '0', '0', '1', '0', '0', '0', '0', '1', '0', '0',
'0', '1', '0', '0', '0', '1', '1', '1', '0', '0', '0', '0', '1', '0', '1', '0', '1', '1', '1', '0', '1',
'0', '0', '0', '0', '0', '0', '0']
```

…

```
K16 = ['0', '0', '0', '1', '0', '1', '1', '0', '0', '0', '0', '1', '0', '1', '0', '0', '0', '1', '1', '0',
'0', '1', '0', '1', '1', '0', '0', '1', '0', '0', '0', '1', '1', '1', '1', '0', '0', '0', '1', '1', '0',
'1', '1', '0', '0', '1', '0', '0']
```

Lines of hexadecimal ciphertext were decrypted iteratively. Ciphertext was converted to a list of string objects of ‘0’ and ‘1’ similar to the key and IV objects. The object was first permutated using IP and sliced in half. The 16 rounds of permutations were run in reverse for decryption. First, the 32-bit right half of the ciphertext was expanded using E to 48-bit:

```
['0', '0', '0', '1', '1', '1', '1', '1', '0', '1', '0', '0', '0', '0', '0', '1', '1', '1', '0', '0', '1',
'0', '1', '0', '1', '1', '1', '0', '1', '1', '0', '0'] =>

['0', '0', '0', '0', '1', '1', '1', '1', '1', '1', '1', '0', '1', '0', '1', '0', '0', '0', '0', '0', '0',
'0', '1', '1', '1', '1', '1', '0', '0', '1', '0', '1', '0', '1', '0', '1', '0', '1', '1', '1', '0', '1',
'0', '1', '1', '0', '0', '0']
```

followed by a conversion to a list of integers and XOR with the key for the round:

```
[0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1,
1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0]
```

The XORed key is permutated by the S Box to 32-bits:

```
['0', '0', '0', '1', '0', '1', '0', '1', '1', '0', '1', '1', '0', '0', '0', '0', '1', '0', '0', '0', '1',
'1', '0', '0', '0', '0', '1', '1', '0', '1', '0', '1']
```

And permutated with P:

```
['0', '0', '0', '1', '0', '1', '1', '1', '0', '0', '0', '0', '0', '0', '0', '0', '0', '1', '0', '0', '1',
'1', '0', '1', '0', '0', '1', '1', '1', '1', '1', '0']
```

The left 32-bits of the ciphertext is XORed with the above:

```
[1, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0]
```

The new left data is swapped with the right data from the start of the iteration for the following round. Following the 16 rounds of swaps, it is permutated by IP-1 and XORed with IV to yield the binary and hexadecimal of the plaintext and converted to plaintext for the first 8 characters:

```
[0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1] =>

5468652050657273 => ['T', 'h', 'e', ' ', 'P', 'e', 'r', 's']
```

The initial ciphertext is then assigned as the IV for the next set of 64-bit ciphertext. This yielded the following plaintext for the passage:

> The Perseids are a prolific meteor shower associated with the comet Swift-
> Tuttle. The Perseids are so-called because the point from which they appear to
> come, called the radiant, lies in the constellation Perseus. The name derives in
> part from the word Perseides, a term found in Greek mythology referring to the
> sons of Perseus.
>
> The stream of debris is called the Perseid cloud and stretches along the orbit
> of the comet Swift-Tuttle. The cloud consists of particles ejected by the comet
> as it travels on its 133-year orbit. Most of the particles have been part of the
> cloud for around a thousand years. However, there is also a relatively young
> filament of dust in the stream that was pulled off the comet in 1865. The rate
> of meteors originating from this filament is much higher than for the older part
> of the stream.

```
The Perseid meteor shower has been observed for about 2,000 years, with the
earliest information on this meteor shower coming from the Far East. Some
Catholics refer to the Perseids as the "tears of St. Lawrence", since 10 August
is the date of that saint's martyrdom.

The shower is visible from mid-July each year, with the peak in activity being
between 9 and 14 August, depending on the particular location of the stream.
During the peak, the rate of meteors reaches 60 or more per hour. They can be
seen all across the sky, but because of the path of Swift-Tuttle's orbit,
Perseids are primarily visible in the northern hemisphere. As with all meteor
showers, the rate is greatest in the pre-dawn hours, since the side of the Earth
nearest to turning into the sun scoops up more meteors as the Earth moves
through space. Most Perseids disappear while at heights above 80 kilometres (50
mi). In 2009, the estimated peak Zenithal Hourly Rate was 173, but fainter
meteors were washed out by a waning gibbous    moon.
```

III. RSA

The Pollard *p*-1 algorithm from the book was implemented to calculate $d = p = 761059198034100157$. This gave $q = \frac{n}{p} = 894843875712616235394832743246282395 63$ and $\phi(n) = (p-1)(q-1) = 68102916241556970634881544571425211463068208887069571672$. The value of a was found by implementing the multiplicative inverse algorithm from the book between $\phi(n)$ and b, $a = 164628369144807846102863394512495811877072171724655621 07$.

For each line of ciphertext, the corresponding plaintext integer was found by $x = y^a \bmod n$ and padded with an additional 0 at the front if the number of digits was odd, for example:

$$045677189638968600661900180808380078374837663767375819 = 491586154035823627790851770627961918208336524240308458 10^a \bmod n$$

Each two characters were used to lookup the plaintext using the mapping matrix to yield the following plaintext for the first string:

```
Hampered by poor visibility
```

And for the entire plaintext:

```
Hampered by poor visibilityand water inside the INS Sindhurakshak submarine thatexploded and sank,
Navy divers on Thursday struggled to locate the 18 trapped personnel on board who are feared killed.

As the hopes for the survival of the sailors in the multiple explosions in the naval dockyard on
early Wednesday receded,Prime Minister Manmohan Singh voiced deep pain at theaccident. Three officers
were among the 18 personnel.

\The trapped personnel have not yet been sighted or recovered,\ a Navy release said, as diving and
salvageoperations continued roundthe clock.

\The diving efforts are hampered by poor visibility inside submarinewhich is filled with water,
extremely restricted access and displacement of mostequipment from their original location,\ the
release said.

The heat of the explosion has melted parts of the internal hull deforming the submarine hatches and
has prevented access to the compartments.

Heavy duty pumps are being used to pumpout the water from the submarine, the release said, adding
there has been large scale ingress of sea water into the submarine due to the explosion.
```

[CSCI 55500] Homework 2
David Xu

\We are deeply pained that we lost the submarine, INS Sindurakshakin an accident yesterday. Eighteen brave sailors are feared to have lost their lives,\ the Prime Minister said in his speech on the 67th Independence Day.

\The accident is all the more painful because the Navy had recently achieved two majorsuccesses in the form of its first nuclear submarine,INS Arihant and the aircraft carrier, INS Vikrant,\ he said.

In one of the worst disasters to have struck the Navy, a series of explosions rocked its submarine INS Sindhurakshak at the dockyard in Mumbai sinking itpartially in the shallow sea.

On Wednesday, Navy chief Joshi Admiral DK Joshi, who accompanied defence minister AK Antony, did not rule out the possibility of asabotage but said that theindicators so far do not support such a theory.

He had also indicated that there was little hope of survival of 18 personnel on boardthe submarine.

The diesel-electric submarine was commissioned into the Indian Navy in 1997 at a cost of around Rs 400 crore and had gone through a Rs 450 crore extensive upgrade in Russia.

The 2300-tonne Kilo class submarine, powered by a combination of diesel generators and electric batteries, had potent weapons package including the anti-ship 'Club' missiles.