Problem 1. RSA Setup

1. For the decrytion in RSA, we have

$$m = C^d \text{ mod } n = m^{ed} \text{ mod } n$$

This equation exists when

$$ed \equiv 1 \text{ mod } \varphi(n)$$

And this uses Euler's theorem, which requires n is coprime with m

2. Let $k = a\varphi(n)$, $a \in N^*$

(a) Given $gcd(m, n) = 1$

$$\begin{aligned} m^k &\equiv (m^{\varphi(n)})^a \text{ mod } n \\ &\equiv 1^a \text{ mod } n \\ &\equiv 1 \text{ mod } n \end{aligned}$$

Thus we have

$$m^k \equiv 1 \text{ mod } p \quad \text{and} \quad m^k \equiv 1 \text{ mod } q$$

(b) Let $\gcd(m, n) = p$, so $\gcd(m/p, q) = 1$

$$\begin{aligned} m^{k+1} &\equiv p \left[\left(\frac{m}{p}\right)^{k+1} \text{ mod } q \right] \text{ mod } n \\ &\equiv p \left[\left(\frac{m}{p}\right)^{a(p-1)\varphi(q)+1} \text{ mod } q \right] \text{ mod } n \\ &\equiv p \cdot \frac{m}{p} \text{ mod } n \\ &\equiv m \text{ mod } n \end{aligned}$$

Thus we have

$$m^{k+1} \equiv m \text{ mod } p \quad \text{and} \quad m^{k+1} \equiv m \text{ mod } q$$

3. (a) Based on

$$ed \equiv 1 \text{ mod } \varphi(n)$$

We know

$$ed = k + 1 \text{ where } k = a\varphi(n)$$

Since for arbitrary m, we have

$$m^{k+1} \equiv m \text{ mod } p \quad \text{and} \quad m^{k+1} \equiv m \text{ mod } q$$

Therefore

$$m^{ed} \equiv m \text{ mod } n$$

(b) It doesn't matter if m and n is coprime, we can always find that $m^{ed} \equiv m \bmod n$ as long as $m < n$. Therefore, it's not necessary for the condition $\gcd(m, n) = 1$.

Problem 2. RSA Decryption

Given $n = 11413$, we can find that $11413 = 101 \times 113$
Then we can find that $\varphi(n) = 112 * 100 = 11200$
Knowing $e = 7467$ and $ed \equiv 1 \mod \varphi(n)$, we have

$$7467 \times 3 \equiv 1 \bmod 11200$$

Therefore $d = 3$, then we can compute m using

$$\begin{aligned} m &\equiv c^d \bmod n \\ m &\equiv 5859^3 \bmod 11413 \\ m &= 1415 \end{aligned}$$

Problem 3. Breaking RSA

1. Based on the encryption and decryption equations, we use

$$m = C^d \bmod n = m^{ed} \bmod n$$

When the d or e is small, the speed for encryption and decryption computation will be faster rather than some large number.

2. Wiener's attack uses the continued fraction method to expose the private key d when $d < \frac{1}{3}N^{\frac{1}{4}}$. The procedure follows as

$$\lambda(N) = lcm(p-1, q-1) = \frac{(p-1)(q-1)}{G} = \frac{\varphi(N)}{G} \text{ where } G = \gcd(p-1, q-1)$$

Since

$$ed \equiv 1 \bmod \lambda(N)$$

There exists an integer K, such that

$$ed = K\lambda(N) + 1$$

$$ed = \frac{K}{G}(p-1)(q-1) + 1$$

Defining $k = \dfrac{K}{\gcd(K, G)}$, $g = \dfrac{G}{\gcd(K, G)}$,

$$ed = \frac{k}{g}(p-1)(q-1) + 1$$

Dividing dpq,

$$\frac{e}{pq} = \frac{k}{dg}(1-\delta), \delta = \frac{p+q-1-\frac{g}{k}}{pq}$$

Since $\dfrac{e}{pq}$ is slightly smaller than $\dfrac{k}{dg}$, assuming $ed > pq$, we have

$$edg = k(p-1)(q-1) + g$$

$$\varphi(n) = (p-1)(q-1) = \frac{g(ed-1)}{k}$$

Then we can use continued fractions to expand $\frac{e}{pq}$, and varify all convergent $\frac{g}{k}$

$$x^2 - ((pq - \varphi(pq)) + 1)x + pq = 0$$

The solutions for the above equation is just p and q, and thus we find the factorization.

3. There are mainly two techniques.
   (a) Choose large public key. Replace e by $e' = e + k\lambda(N)$ for large k, s.t $e' > N^{\frac{3}{2}}$, then Wiener's Attack cannot be applied even d is small.
   (b) Using CRT, choose d satisfies $d_p = d \bmod p-1$ and $d_q = d \bmod q-1$, then let

   $$M_p \equiv C^{d_p} \text{ and } M_q \equiv C^{d_q}$$

   Find M that

   $$M \equiv M_p \bmod p \text{ and } M \equiv M_q \bmod q$$

   Since $d \bmod \lambda(N)$ is large, Wiener's attack cannot be applied

4. Given $n = 317940011$ and $e = 77537081$, we first try to find continued fraction of $\frac{e}{n}$

```
#include <cstdint>
#include <iostream>
#include <cmath>

using namespace std;

void continued_frac(uint64_t p, uint64_t q, double limit){
    uint64_t h1 = 1, k1 = 0, h2 = 0, k2 = 1;
    size_t i = 0;
    while (q >= 1){
        auto d = p / q;
        auto h = d * h1 + h2;
        auto k = d * k1 + k2;
        if (k > limit){
            break;
        }
```

```
        cout << " k = " << h << ", d = " << k << endl;

        h2 = h1;
        h1 = h;
        k2 = k1;
        k1 = k;

        d = p % q;
        p = q;
        q = d;
    }
}

int main(){
    uint64_t n = 317940011;
    uint64_t e = 77537081;
    double limit = pow(n, 0.25)/3;
    continued_frac(e, n, limit);

    return 0;
```

The applicable values for Wiener's attack are following

```
k = 0, d = 1
k = 1, d = 4
k = 9, d = 37
k = 10, d = 41
```

Then we need to verify $\phi(n)$ such that $(n-\varphi(n)+1)^2-4n$ is a square and thus we have solutions for p,q. We find that when $k_0=10$, $d=41$,

$$\phi(n)=\frac{ed-1}{k}=317902032$$

$$(n-\varphi(n)+1)^2-4n=170720356=13066^2$$

$$p=\frac{37980+13066}{2}=25523, \text{ and } q=\frac{37980-13066}{2}=12457$$

Therefore,

$$n=317940011=25523\times 12457$$

Problem 4. Programming

Source code is uploaded. Please see README.

Problem 5. Simple questions

1. For CCA, if we are given a ciphertext c which encrypts a message m, then we can choose $c' \equiv c \cdot 2^e \bmod n$, when the owner decrypt the c' use d, he'll get $(2m)^e \bmod n$ since $c = m^e$.

2. No. The hard problem to solve in RSA is a factorization problem. As long as n could be factorized, it's simple to derive d, and the decryption procedure is the same regarding multiple layers of encryption.

3. For $n = 642401$, knowing $516107^2 \equiv 7 \bmod n$ and $187722^2 \equiv 4 \cdot 7 \bmod n$. We have

$$4 \cdot 516107^2 \equiv 4 \cdot 7 \bmod n$$

Then minus two equations, we have

$$4 \cdot 516107^2 - 187722^2 \equiv 0 \mod n$$

$$(2 \cdot 516107 - 187722)(2 \cdot 516107 + 187722) \equiv 0 \mod n$$
$$1219936 \cdot 844492 \equiv 0 \mod n$$
$$2^5 \times 38123 \cdot 2^2 \times 2123649 \equiv 0 \mod n$$
$$2^5 \times 67 \times 569 \cdot 2^2 \times 11 \times 171 \times 1129 \equiv 0 \mod n$$

Since 642401 must be the product of the factors list above, we can find that

$$n = 642401 = 569 \times 1129$$

4. Consider $n = p \cdot q \cdot r$, where p, q, r are large prime numbers, then

$$\varphi(n) = (p-1)(q-1)(r-1)$$

As with textbook RSA encrytion and decryption method, we should have

$$ed \equiv 1 \mod \varphi(n)$$

$$c^d \equiv m^{ed} \equiv m^{\varphi(n)+1} \equiv m \mod n$$

Since we have three prime factors for n, the length for each prime factor is smaller, and it's much easier to factorize n, which lower the security of the RSA.

5. 97 is a prime, and $(97-1) = 96 = 2^5 \times 3$, then $q = 2$ or 3.
Thus generator x should satisfy

$$x^{32} \neq 1 \mod 97 \quad \text{and} \quad x^{48} \neq 1 \mod 97$$

We find 5 satisfies all requirements, thus the smallest generator of $U(Z/97Z)$ is 5.

6. (a) 137 is a prime, and $101 - 1 = 100 = 2^2 \times 5^2$, thus $q = 2$ or 5.

$$\begin{aligned} 2^{100/2} &\equiv (2^{10})^5 \mod 101 \\ &\equiv 14^5 \mod 101 \\ &\equiv 100 \mod 101 \\ 2^{100/5} &\equiv (2^{10})^2 \mod 101 \\ &\equiv 14^2 \mod 101 \\ &\equiv 95 \mod 101 \end{aligned}$$

Since $2^{50} \not\equiv 1 \mod 101$ and $2^{20} \not\equiv 1 \mod 101$, 2 is a generator of G.

(b) Given $\log_2 3 = 69$, knowing $\log_2 2 = 1$

$$\begin{aligned} \log_2 24 &= \log_2(3 \times 2^3) \\ &= \log_2 3 + 3\log_2 2 \\ &= 69 + 3 = 72 \end{aligned}$$

(c) Given $\log_2 5 = 24$

$$\begin{aligned} \log_2 24 &= \log_2 125 \\ &= 3\log_2 5 \\ &= 3 \times 24 = 72 \end{aligned}$$

7. Knowing $3^6 \equiv 44 \bmod 137$, and $3^{10} \equiv 2 \bmod 137$, since $(137 - 1) = 136 = 2^3 \times 17$, $q = 2$ or 17.

$$\begin{aligned} 3^{136/2} &\equiv 3^5 \cdot (3^7)^9 \mod 137 \\ &\equiv 243 \cdot (-5)^9 \mod 137 \\ &\equiv 106 \cdot 12^3 \mod 137 \\ &\equiv 106 \cdot 7 \cdot 12 \mod 137 \\ &\equiv 136 \mod 137 \\ 3^{136/17} &\equiv 3^8 \mod 137 \\ &\equiv 3 \cdot -5 \mod 137 \\ &\equiv 122 \mod 137 \end{aligned}$$

Since $3^{68} \not\equiv 1 \mod 137$ and $3^8 \not\equiv 1 \mod 137$, 3 is a generator of $U(Z/137Z)$. Since

$$\log_3 11 = \log_3 44 - 2\log_3 2 = -14$$

We have

$$3^{-14} \equiv 11 \bmod 137$$

$$3^{136} \equiv 1 \bmod 137$$

Thus $x = 122$.

8. (a) $6^5 = 7776 \equiv 10 \mod 11$, thus $6^5 = 10$ in $U(Z/11Z)$

   (b) 11 is a prime, and $(11-1) = 10 = 2 \times 5$, $q = 2$ and 5

   $$2^{10/2} \equiv 10 \mod 11$$

   $$2^{10/5} \equiv 4 \mod 11$$

   Since $2^5 \not\equiv 1 \mod 11$ and $2^2 \not\equiv 1 \mod 11$, 2 is a generator of G.

   (c) Given $2^x \equiv 6 \mod 11$, using $6^5 = 10$ mod 11, we have

   $$\begin{aligned}(2^x)^5 &\equiv 6^5 \mod 11\\ (2^5)^x &\equiv 10 \mod 11\\ (-1)^x &\equiv -1 \mod 11\end{aligned}$$

   Thus we find that x should be odd.

Problem 6. DLP

1. Given $3^x \equiv 2$ mod 65537, we know $2^{16} = 65536$

   $$\begin{aligned}3^{16x} = 2^{16} &\equiv -1 \text{ mod } 65537\\ 3^{32x} &\equiv 1 \text{ mod } 65537\end{aligned}$$

   Since we know 3 is generator for $U(Z/65537Z)$, we have

   $$3^{65536} \equiv 1 \text{ mod } 65537$$

   Thus $65536 \mid 32x$, we have $2048 \mid x$.

   Since $65536 \nmid 16x$, we have $4096 \nmid x$.

2. Given $2048 \mid x$ and $4096 \nmid x$, let $x = 2048k$, where k must be odd. Since $0 \leq k < ord_U(3) = 32$, k should be less than 32, therefore there are 16 possible choices.

   To determine x, first we determine two boundary values

   $$3^{2048 \cdot 1} \equiv -8 \mod 65537$$

   $$3^{2048 \cdot 31} \equiv 8192 \mod 65537$$

   And then we find that $2 = 8192/(-8)^4$, thus

   $$3^{2048 \cdot (31-4)} \equiv 2 \mod 65537$$

   Therefore $x = 2048 \cdot 27 = 55296$.

3. Yes. According to Pohlig-Hellman algorithm, assume $x = 2^0 + x_1 2^1 + \cdots + x_{15} 2^{15}$, since $x \mid 2048$ and $x \nmid 4096$, we can simply as

$$x = 2^{11} + x_{12}2^{12} + x_{13}2^{13} + x_{14}2^{14} + x_{15}2^{15}$$

To find x_{12}, we have

$$\begin{aligned} 3^{2^{11}} \cdot 3^{2^{11}(x_{12}2^{12}+x_{13}2^{13}+x_{14}2^{14}+x_{15}2^{15})} &\equiv 2 \text{ mod n} \\ 3^{2^{11}(x_{12}2^{1}+x_{13}2^{2}+x_{14}2^{3}+x_{15}2^{4})} &\equiv 2 \cdot 3^{-2^{11}} \text{ mod n} \\ &\equiv 2 \cdot (-2^3)^{-1} \text{ mod n} \\ &\equiv 2^{14} \text{ mod n} \end{aligned}$$

Applying Fermat's little theorem, we have

$$(3^{2^{11}(x_{12}2^{1}+x_{13}2^{2}+x_{14}2^{3}+x_{15}2^{4})})^8 \equiv (2^{14})^8 \equiv -1 \text{ mod } 65537$$

$$(3^{2^{15}})^{x_{12}} = (-1)^{x_{12}} = -1 \text{ mod } 65537$$

$$x_{12} = 1$$

Similarly, we find $x_{13} = 0, x_{14} = 1, x_{15} = 1$

Thus, $x = 2^{11} + 2^{12} + 2^{14} + 2^{15} = 55296$.

4. Because 65537 is a special prime that can be expressed as $p^k + 1$, where $p = 2$ and $k = 16$. In order to find x value for this type of numbers, we just need to find a generator α. Since $c^{2k} \equiv p^{2k} \equiv 1 \mod p^k + 1$, we have $p^k/2k$ divides x while p^k/k not. According to problem 2, there are only k possible choices left for x, which is not secure under a cryptography context.