Chapter 4

Exponentiation

Modular exponentiation, e.g., the operation $a^k \,\%\, n$, often plays a major part in modern practice of cryptographical procedures. Topics such as the RSA cryptosystem (Section 4.3) have only recently—some twenty years ago—become almost a standard motivational chapter in a typical number theory course. On the theoretical side, modular exponentiation begins with an elegant theorem of Fermat and its generalization by Euler.

4.1 Fermat's Theorem and Euler's Function

Recall that a complete residue system modulo n is a set of representatives of the residue classes modulo n, exactly one number for each class. The following lemma will lead to the theorem of Fermat.

Lemma 4.1. Assume that $\gcd(a,n) = 1$. Then $\{r_1, r_2, \ldots, r_n\}$ is a complete residue system modulo n if and only if $\{ar_1, ar_2, \ldots, ar_n\}$ is also a complete residue system modulo n.

Proof. By Proposition 3.3, $ar_j \equiv ar_k \pmod{n}$ implies $r_j \equiv r_k \pmod{n}$, since $\gcd(a,n) = 1$. In that case, $\{ar_1, ar_2, \ldots, ar_n\}$ represents distinct congruence classes modulo n if and only if $\{r_1, r_2, \ldots, r_n\}$ also represents distinct congruence classes modulo n. $\bigtriangledown$

Example. We illustrate Lemma 4.1 using $a = 4$ and $n = 9$. Note that $\gcd(4,9) = 1$. Multiply by 4 each number in $\{0,1,2,3,4,5,6,7,8\}$, which is a complete residue system modulo 9. We obtain $\{0,4,8,12,16,20,24,28,32\}$. We check this finding by taking residues mod 9, keeping the same order, and get $\{0,4,8,3,7,2,6,1,5\}$. It is again a complete residue system.

Theorem 4.2 (Fermat's Little Theorem[1]). If $p \nmid a$ then $a^{p-1} \equiv 1 \pmod{p}$, where p is any prime number.

Proof. By Lemma 4.1, the numbers 0, a, $2a$, ... , $(p-1)a$ form a complete residue system modulo p, hence their residues mod p are 0, 1, 2, ... , $p-1$, but not necessarily in this order. Leaving 0 out, these two groups of numbers make up the following congruence.

$$a \times 2a \times \cdots \times (p-1)a \;\equiv\; 1 \times 2 \times \cdots \times (p-1) \pmod{p}$$

Wilson's theorem gives us $-a^{p-1} \equiv -1 \pmod{p}$, thus the result. $\triangledown$

For example, with prime $p = 101$ and $a = 2$, we have $2^{100} \,\%\, 101 = 1$. Beware, however, sometimes a composite may behave likewise, for instance $29^{34} \equiv 1 \pmod{35}$. Hence, unlike Wilson's theorem, Fermat's little theorem is not a primality criterion. Nevertheless, Theorem 4.2 is still the basis for many modern primality testing algorithms, some of which we will see in Section 9.1.

Exercise 4.1. Investigate true or false.

a) If $a^n \equiv a \pmod{n}$ then n is a prime.
b) If $a^n \not\equiv a \pmod{n}$ then n is composite.
c) If $a \equiv b \pmod{n}$ then $a^k \equiv b^k \pmod{n}$.
d) If $j \equiv k \pmod{n}$ then $a^j \equiv a^k \pmod{n}$.

Exercise 4.2. Show that Fermat's little theorem can well be rephrased as follows. *If p is a prime, then $a^p \equiv a \pmod{p}$ for any integer a.*

We aim next to find a congruence property similar to that in Theorem 4.2, but for composite moduli. The first step in that direction is the introduction of the phi function, due to Euler.

Definition. The *Euler phi function* $\phi(n)$ is the number of positive integers up to n which are relatively prime to n. For example, in the range from 1 to 12, only 1, 5, 7, and 11 are relatively prime to 12. Therefore $\phi(12) = 4$. Similarly, $\phi(11) = 10$.

Exercise 4.3. Evaluate $\phi(n)$ for the following values of n.

a) $n = 13$
b) $n = 14$
c) $n = 15$
d) $n = 16$

[1] Little, in comparison to his bigger, then unproved *last theorem,* which states that the diophantine equation $x^n + y^n = z^n$ has no nontrivial solution for $n \geq 3$.

Exercise 4.4. Show that $\phi(p) = p - 1$ for any prime p. Conversely, prove that only prime numbers can satisfy the property $\phi(n) = n - 1$.

Note that in any complete residue system modulo n, the number of elements which are relatively prime to n is invariably $\phi(n)$. This fact is a consequence of Theorem 1.4, and it allows us to give the next definition.

Definition. A *reduced residue system* modulo n is a subset of a complete residue system modulo n, consisting of the $\phi(n)$ numbers relatively prime to n. For example, a reduced residue system modulo 9 could be $\{1, 2, 4, 5, 7, 8\}$, or $\{\pm1, \pm2, \pm4\}$, or another, but each one will have $\phi(9) = 6$ elements.

Exercise 4.5. Find a reduced residue system modulo n, consisting of only prime numbers—Theorem 2.8 foresees infinitely many such systems.

a) $n = 12$
b) $n = 13$
c) $n = 14$
d) $n = 15$
e) $n = 24$

One more lemma and we are ready to prove Euler's theorem, which generalizes Fermat's little theorem to arbitrary moduli, not necessarily prime.

Lemma 4.3. Suppose that $\gcd(a, n) = 1$. Then $\{r_1, r_2, \ldots, r_{\phi(n)}\}$ is a reduced residue system modulo n if and only if $\{ar_1, ar_2, \ldots, ar_{\phi(n)}\}$ is also a reduced residue system modulo n.

Proof. As in the proof of Lemma 4.1, either both sets represent distinct congruence classes or neither does, says Proposition 3.3. Moreover, by Theorem 1.8(3), $\gcd(r_i, n) = 1$ if and only if $\gcd(ar_i, n) = 1$. This establishes the claim. ▽

Example. Let us take $\{1, 2, 4, 5, 7, 8\}$ as a reduced residue system modulo 9. Multiplying each number by 4 results in $\{4, 8, 16, 20, 28, 32\}$ with residues mod 9, in this order, $\{4, 8, 7, 2, 1, 5\}$. Hence, we get another reduced residue system modulo 9. Note the fact that $\gcd(4, 9) = 1$.

Theorem 4.4 (Euler's Theorem)**.** If $\gcd(a, n) = 1$ then $a^{\phi(n)} \equiv 1 \pmod{n}$, for any positive integer n.

Proof. If $\gcd(a, n) = 1$ then by Lemma 4.3, we may choose a reduced residue system modulo n which looks like $\{r_1, r_2, \ldots, r_{\phi(n)}\}$, and another one, $\{ar_1, ar_2, \ldots, ar_{\phi(n)}\}$. These elements form the following congruence.

$$a^{\phi(n)} r_1 r_2 \cdots r_{\phi(n)} \equiv r_1 r_2 \cdots r_{\phi(n)} \pmod{n}$$

Then simply cancel the r_i's off both sides, as allowed by Proposition 3.3. ▽

Note that when n is a prime number, Euler's theorem and Fermat's little theorem coincide. Also, the structures of the two proofs are so similar that some lecturers would rather present Euler's theorem first before stating Fermat's little theorem as a direct corollary.

Exercise 4.6. If $a^k \equiv 1 \pmod{n}$ for some $k > 0$, show that $\gcd(a, n) = 1$.

For practical purposes, Euler's theorem is not of much use until we learn a more feasible way to evaluate $\phi(n)$. We devote the rest of the section solely with this goal in mind.

Theorem 4.5. If $\gcd(m, n) = 1$ then $\phi(mn) = \phi(m)\phi(n)$.

Proof. Let M, N, and MN be reduced residue systems modulo m, n, and mn, respectively. Also denote by $M \times N$ the set consisting of the elements (c, d) with $c \in M$ and $d \in N$. We shall provide a one-to-one correspondence between $M \times N$ and MN, thereby showing that $\phi(m)\phi(n) = \phi(mn)$.

Pick an element $a \in MN$. We have $\gcd(a, mn) = 1$, thus $\gcd(a, m) = 1$ and $\gcd(a, n) = 1$. Since M and N are reduced residue systems, there exists a unique pair $(c, d) \in M \times N$ such that $a \equiv c \pmod{m}$ and $a \equiv d \pmod{n}$. Conversely, given a pair of congruences $x \equiv c \pmod{m}$ and $x \equiv d \pmod{n}$ with $(c, d) \in M \times N$, by the Chinese remainder theorem (Theorem 3.10), $x = a$ is the unique element in MN which solves the system. This establishes the one-to-one correspondence between the two sets. $\triangledown$

As a consequence of Theorem 4.5, we are now able to evaluate $\phi(n)$ with ease, provided that the factorization of n has been established.[2]

Proposition 4.6. If n factors into prime powers, written $n = \prod p_i^{e_i}$, then

$$\phi(n) = \prod \phi(p_i^{e_i}) = \prod (p_i^{e_i} - p_i^{e_i - 1}) = \prod p_i^{e_i - 1}(p_i - 1)$$

Proof. Theorem 4.5 permits us to prove only that $\phi(p^e) = p^e - p^{e-1}$. Simply recall that $\phi(p^e)$ is the number of integers from 1 to p^e which are relatively prime to p^e. Since p is the only prime divisor of p^e, then $\phi(p^e)$ equals p^e minus the number of multiples of p. And the multiples of p are p, $2p$, $3p$, ... , up to $(p^{e-1})p$—exactly p^{e-1} of them. $\triangledown$

Example. To evaluate $\phi(600)$, we first factor $600 = 2^3 \times 3 \times 5^2$, and then we apply Proposition 4.6.

$$\phi(600) = \phi(2^3)\phi(3)\phi(5^2) = (2^3 - 2^2)(3 - 1)(5^2 - 5) = 4 \times 2 \times 20 = 160$$

[2] But yes, we did say that factoring is slow...

Exercise 4.7. Evaluate $\phi(n)$ for the following values of n.

a) $n = 240$
b) $n = 625$
c) $n = 1024$
d) $n = 4800$
e) $n = 19392$

Exercise 4.8. Find all positive integers n, such that $\phi(n) = 4$.

Exercise 4.9. Prove the following facts about $\phi(n)$.

a) If n is odd then $\phi(2n) = \phi(n)$.
b) If n is even then $\phi(2n) = 2\phi(n)$.
c) If $n > 2$ then $\phi(n)$ is even.
d) If $d \mid n$ then $\phi(d) \mid \phi(n)$.

Exercise 4.10. Another property of the phi function is that $\sum \phi(d) = n$, where the sum is taken over the range $1 \le d \le n$ such that $d \mid n$. Prove it.

4.2 Computing Large Powers

In cryptographical applications, we mentioned earlier, it is often necessary to perform the exponentiation $a^k \,\%\, n$, and that with a very large value of k. One obvious way to compute a^k is multiplying a by itself k times, where the partial product in each step can be reduced mod n in order to keep the size small—doing so will have no effect on the final answer, and the following exercise asks for its justification.

Exercise 4.11. Show that $a^2 \,\%\, n = (a \,\%\, n)^2 \,\%\, n$. More generally, prove that $ab \,\%\, n = (a \,\%\, n)(b \,\%\, n) \,\%\, n$.

Moreover, it can be assumed that $a < n$, since $a^k \,\%\, n = (a \,\%\, n)^k \,\%\, n$. If $\phi(n)$ is known, or easily computed, and if $\gcd(a, n) = 1$, then by Euler's theorem we have $a^k \,\%\, n = 1$ whenever $\phi(n) \mid k$. More generally, for arbitrary k, we have

$$a^k \,\%\, n = a^{k \,\%\, \phi(n)} \,\%\, n$$

For small values of n, this knowledge may come in handy.[3]

Example. Say, we have to compute $1234^{5678} \,\%\, 11$. We note that $1234 \,\%\, 11 = 2$, relatively prime to 11. Also, $\phi(11) = 10$ and $5678 \,\%\, 10 = 8$. Therefore, $1234^{5678} \,\%\, 11 = 2^8 \,\%\, 11 = 256 \,\%\, 11 = 3$.

Exercise 4.12. Compute these residues with the help of Euler's theorem.

[3] To do with $a^k \,\%\, n$, reduce the base a mod n and the power k mod $\phi(n)$.

a) $83^{3418} \,\%\, 24$
b) $49^{2324} \,\%\, 41$
c) $3337^{3331} \,\%\, 64$
d) $2234^{2600} \,\%\, 97$
e) $3294^{3845} \,\%\, 143$

EXERCISE 4.13. Find the unit digit upon computing 123^{45678}.

Euler's theorem does not apply, however, when $\gcd(a, n) > 1$. Besides, evaluating $\phi(n)$ involves factoring, which we would rather avoid. This will not matter anyhow once we have learned the so-called *successive squaring algorithm*. The idea is based on the fact that every positive integer is the sum of powers of two.

EXERCISE 4.14. For $k \geq 0$, show that there is a unique way to write $k = \sum_{i\geq 0} b_i \times 2^i$, where $b_i \in \{0, 1\}$. In particular, $b_i = 0$ for all $i > \log_2 k$.

Not only does successive squaring reduce computation time significantly,[4] but this algorithm also works regardless of $\gcd(a, n)$. To understand successive squaring algorithm, when you've seen one example, you've seen them all.

Example (Successive Squaring Algorithm). Compute $23^{106} \,\%\, 97$. We have $106 = 64+32+8+2 = 2^6+2^5+2^3+2^1$, and so $23^{106} = 23^{64} \times 23^{32} \times 23^8 \times 23^2$. In the next step we successively square the base number 23, thus the name.

$$\begin{aligned}
23^2 \,\%\, 97 &= 44 \\
23^4 \,\%\, 97 = 44^2 \,\%\, 97 &= 93 \\
23^8 \,\%\, 97 = 93^2 \,\%\, 97 &= 16 \\
23^{16} \,\%\, 97 = 16^2 \,\%\, 97 &= 62 \\
23^{32} \,\%\, 97 = 62^2 \,\%\, 97 &= 61 \\
23^{64} \,\%\, 97 = 61^2 \,\%\, 97 &= 35
\end{aligned}$$

Hence, $23^{106} \,\%\, 97 = (35 \times 61 \times 16 \times 44) \,\%\, 97 = 25$.

EXERCISE 4.15. Use successive squaring to compute these residues.

a) $3^{57} \,\%\, 20$
b) $25^{99} \,\%\, 79$
c) $47^{250} \,\%\, 200$
d) $5^{1434} \,\%\, 307$
e) $25^{1434} \,\%\, 309$

[4] It uses only $O(\log k)$ instead of k multiplications.

Exercise 4.16. Find the two right-most digits of the number 123^{45678}.

Concluding this section, we will next, for merely theoretical amusement, investigate to what extent Euler's theorem fails when $\gcd(a, n) > 1$.[5]

Let a and n be arbitrary positive integers. Set $n_0 = n$ and $d_0 = \gcd(a, n)$. Then for $k \geq 1$, we define n_k and d_k recursively by

$$n_k = \frac{n_{k-1}}{d_{k-1}} \quad \text{and} \quad d_k = \gcd(a, n_k)$$

It is an easy exercise to show that both sequences, n_k and d_k, are strictly decreasing with $n_{k+1} \mid n_k$ and $d_{k+1} \mid d_k$, until they become stationary when $d_k = 1$. Let L be the least integer for which $d_L = 1$. The following result can be viewed as an extension of Euler's theorem.

Theorem 4.7. *Let a and n be arbitrary positive integers with n_k, d_k, and L defined as above. Then the first repeated term in the sequence $\{a^k \,\%\, n\}$ occurs at $k = L$. Moreover,*

$$a^{\phi(n_L)} a^L \equiv a^L \pmod{n}$$

with which Euler's theorem coincides in the case $L = 0$.

Proof. Suppose that $a^i \equiv a^j \pmod{n}$ with $i > j$. This is equivalent, through division by d_0, to the congruence $a^i/d_0 \equiv a^j/d_0 \pmod{n_1}$. Corollary 1.7 tells us that $\gcd(a/d_0, n_1) = 1$, hence by Proposition 3.3, the congruence is again equivalent to $a^{i-1} \equiv a^{j-1} \pmod{n_1}$. Repeat this process of substitutions j times, and we arrive at $a^{i-j} \equiv 1 \pmod{n_j}$. For one thing, this implies that $\gcd(a, n_j) = 1$, thus $j \geq L$. In particular, with $i = \phi(n_L) + L$ and $j = L$, the congruence $a^{\phi(n_L)} a^L \equiv a^L \pmod{n}$ is equivalent to $a^{\phi(n_L)} \equiv 1 \pmod{n_L}$, which holds because of Euler's theorem. ▽

Example. We illustrate with $a = 2^3 \times 3^2 \times 5$ and $n = 2^7 \times 3 \times 5^2 \times 7$,

$$\begin{aligned}
n_0 &= n & &= 2^7 \times 3 \times 5^2 \times 7 & d_0 &= \gcd(a, n_0) = 2^3 \times 3 \times 5 \\
n_1 &= n_0/d_0 & &= 2^4 \times 5 \times 7 & d_1 &= \gcd(a, n_1) = 2^3 \times 5 \\
n_2 &= n_1/d_1 & &= 2 \times 7 & d_2 &= \gcd(a, n_2) = 2 \\
n_3 &= n_2/d_2 & &= 7 & d_3 &= \gcd(a, n_3) = 1
\end{aligned}$$

Meanwhile, the sequence $a^k \,\%\, n$ generates the following numbers.

$$360,\ 62400,\ 19200,\ 57600,\ 38400,\ 48000,\ 9600,\ 28800,\ 19200,\ \ldots$$

[5] The uninterested reader may instead skip forward to the next section, or even to the next chapter, without violating the logical chronology of the text.

Note that $a^3 \,\%\, n = 19200$ is the first repeated term, which corresponds to $L = 3$. Moreover, that $\phi(n_3) = \phi(7) = 6$ is reflected in the fact that beyond the third term, the sequence repeats itself every six terms. In general, however, the length of the periodicity will be a divisor of $\phi(n_L)$, not necessarily equal to it.

Suppose, for instance, we further wish to compute $a^{8888} \,\%\, n$. Although $8888 \,\%\, 6 = 2$, here we have $a^{8888} \not\equiv a^2 \pmod{n}$ since $2 < L$. Instead, we have $a^{8888} = a^{8885}a^3 \equiv a^5a^3 = a^8 \pmod{n}$. That is, $2^{8888} \,\%\, n = 28800$.

Exercise 4.17. Compute these residues following the above example.

a) $2^{456} \,\%\, 10$
b) $10^{456} \,\%\, 36$
c) $42^{654} \,\%\, 88$
d) $126^{9999} \,\%\, 432$
e) $385^{3422} \,\%\, 900$

Exercise 4.18. Writing $n = \prod p_i^{e_i}$ and $a = \prod p_i^{f_i}$ with $e_i, f_i \geq 0$, show that

$$n_L = \prod_{p_i \nmid a} p_i^{e_i} \quad \text{and} \quad L = \max_{f_i \neq 0} \left\lceil \frac{e_i}{f_i} \right\rceil$$

(See Exercises 1.5 and 2.10 for the ceiling and max notation, respectively.)

Corollary 4.8. If n has no repeated prime factors, then $a^{\phi(n)}a \equiv a \pmod{n}$.

Proof. The condition implies that $L \leq 1$. Either $L = 0$, which is trivially Euler's theorem, or $L = 1$ and $a^{\phi(n_1)}a \equiv a \pmod{n}$. But $\phi(n_1) \mid \phi(n)$ since $n_1 \mid n$, (Exercise 4.9(d)) thus the result. ▽

Exercise 4.19. Prove Corollary 4.8 again without relying on Theorem 4.7, this time using the Chinese remainder theorem.

4.3 The RSA Cryptosystem

If, instead of computing $a^k \,\%\, n$, we are given its value and asked to retrieve a, then what we are facing is the more difficult problem of modular root extraction. Under some relatively prime conditions, the problem is not difficult to solve, at least theoretically. The following result is in fact a key principle employed in the RSA cryptosystem, soon to be introduced.

Theorem 4.9. If both $\gcd(s, n)$ and $\gcd(e, \phi(n))$ equal 1, then the congruence $x^e \equiv s \pmod{n}$ has a unique root modulo n given by $x \equiv s^d \pmod{n}$, where $d \equiv e^{-1} \pmod{\phi(n)}$.

Proof. We will prove that the two congruences modulo n are equivalent. Modular inverse theorem (Corollary 3.6) guarantees the existence, as well as uniqueness, of d modulo $\phi(n)$. For one and any such d, say $de = 1+\phi(n)h$ for some integer h, the congruence $x \equiv s^d \pmod{n}$ implies

$$x^e \equiv s^{de} = s^{1+\phi(n)h} = s(s^{\phi(n)})^h \equiv s \pmod{n}$$

by way of Euler's theorem. Conversely, the congruence $x^e \equiv s \pmod{n}$, when raised to the power d, will give back $x \equiv s^d \pmod{n}$. $\bigtriangledown$

Exercise 4.20. Solve for x.

a) $x^7 \equiv 12 \pmod{13}$
b) $x^{13} \equiv 5 \pmod{32}$
c) $x^{39} \equiv 5 \pmod{121}$
d) $x^{121} \equiv 30 \pmod{899}$
e) $x^{239} \equiv 23 \pmod{2005}$

Now sensitive messages, when transfered over electronic media such as the internet, may need to be encrypted, i.e., changed into a secret code in such a way that only the intended receiver who has the secret key is able to decrypt it. It is common that alphabetical characters are converted numerically, for instance according to their ASCII values, before they are encrypted. Hence, the coded message can be treated as integer strings.

For this purpose, the *RSA cryptosystem*[6] provides an encryption and decryption algorithm which is widely employed today. In practice, the encryption key may be made public, and doing so will not risk the security of the system. This feature is a characteristic of the so-called *public-key cryptosystem.*

How does it work? Let's say, the two communicating parties are represented by Alia and Bob. Alia selects two distinct primes p and q which are very large, perhaps of a hundred digits each. She computes $n = pq$ and $\phi(n) = (p-1)(q-1)$. Next Alia determines a number e, less than and relatively prime to $\phi(n)$, which will serve as her *encryption key.* As for her *decryption key,* Alia computes the number $d < \phi(n)$, which satisfies $de \,\%\, \phi(n) = 1$.

When all is ready, Alia gives to Bob the pair (n, e) and keeps the rest secret. In the future, whenever Bob wants to send a message (integer) $m < n$ to Alia, he encrypts m to $s = m^e \,\%\, n$, and sends s instead. Upon receiving s, Alia is able to retrieve m, knowing that $s^d \,\%\, n = m$ by Theorem 4.9.

Why does this work? First of all, there are plenty of primes 100-digits long. In fact there are $\pi(10^{100}) - \pi(10^{99})$ such primes, roughly 3.9×10^{97} of

[6] Named after Rivest, Shamir, and Adleman, who patented it in 1983.

them, and they are not too hard to find using primality testing algorithms available today. Secondly, determining e or d is not too hard for Alia with the help of the Euclidean algorithm. Neither it is hard for Bob to encrypt $s = m^e \,\%\, n$, nor for Alia to decrypt $s^d \,\%\, n = m$, using successive squaring algorithm.

But what if a bad guy, represented by Cobra, intercepts the secret message s, together with e and n? Well, d is yet to be found in order to read the message and for that, the factors p and q will be needed in computing $\phi(n)$. Woe to him, n has over 200 digits, and factoring a large integer this size will take a lifetime on today's fastest computer!

Example. By way of illustration, Alia chooses $n = 19 \times 53 = 1007$, where $\phi(n) = 18 \times 52 = 936$. She also selects her encryption key $e = 5$, which is relatively prime to 936. After working it out shortly using the extended Euclidean algorithm, Alia finds the inverse, $d = 749$, to be the right decryption key, checking it again that $(749 \times 5) \,\%\, 936 = 1$. Then she proceeds to send $n = 1007$ and $e = 5$ to Bob, say, via email.

Later, Bob wishes to send the message `SOS` to Alia. Using a very naive ASCII substitutions, i.e., 65 = `A`, 66 = `B`, ... , 90 = `Z`, the intended message is coded as 837983. To make $m < 1007$, Bob cuts up this string into blocks of 3 digits, in this case, 837; 983. He then sends to Alia the two values of s, in this order,

$$837^5 \,\%\, 1007 = 970 \quad \text{and} \quad 983^5 \,\%\, 1007 = 732$$

Whereas, upon receiving, Alia decrypts these two numbers,

$$970^{749} \,\%\, 1007 = 837 \quad \text{and} \quad 732^{749} \,\%\, 1007 = 983$$

At last, Alia reunites these results back into a single string and reverses the ASCII conversion, to be able to read the urgent message from Bob.

In real practice, of course, n is a much larger integer.

Exercise 4.21. In this RSA exercise, Alia picks $n = 127 \times 79 = 10033$ and $e = 17$.

a) What is her decryption key d?
b) Wanting to say `HI`, what does Bob send to her?
c) Verify that Alia does get this greeting correctly.
d) Another time she receives $s = 8411$. What is the intended message?

Theorem 4.9 assumes, in the context of RSA, that $\gcd(s, n) = 1$. In practice, however, the encrypted message s may fail to be relatively prime to n, although the probability of such coincidence is extremely small as n is a very large number with only two prime factors. Nevertheless, as an exercise we can prove that the decryption algorithm will anyhow return the correct message m.

Exercise 4.22. Suppose that $\gcd(s, n) > 1$. Show that anyway $s^d \,\%\, n = m$.

RSA works under a crucial assumption that it is hard to evaluate $\phi(n)$ without factoring n. Knowing p and q, of course, gives $\phi(n) = (p-1)(q-1)$. Conversely, knowing $\phi(n)$ will easily lead to the discovery of p and q, since they are the two zeros of the quadratic polynomial

$$x^2-(n-\phi(n)+1)x+n = x^2-(pq-(p-1)(q-1)+1)x+pq = (x-p)(x-q)$$

We can then say that evaluating $\phi(n)$ is no less difficult than factoring n.

Example. Suppose $n = 1007$ and $\phi(n) = 936$, as before. Knowing only these two values, we look for the zeros of $x^2 - (1007 - 936 + 1)x + 1007 = x^2 - 72x + 1007$, via the familiar *quadratic formula,*

$$x = \frac{72 \pm \sqrt{72^2 - 4 \times 1007}}{2} = 36 \pm \frac{\sqrt{1156}}{2} = 36 \pm 17$$

Thus, we rediscover, $1007 = (36 + 17)(36 - 17) = 53 \times 19$.

Exercise 4.23. Given $n = pq$ and $\phi(n)$, find p and q.

a) $\phi(209) = 180$
b) $\phi(2231) = 2112$
c) $\phi(11371) = 11152$
d) $\phi(147911) = 147000$

The RSA Laboratories used to post a list of factoring challenge at their site `http://www.rsa.com/rsalabs/`. One of the challenge numbers was the following 232-digit composite, labeled RSA-768, which was worth US$50,000 before the offer expired in 2007.

$$\begin{aligned} n \quad = \quad & 1230186684530117755130494958384962720772853569595\,3 \\ & 3479219732245215172640050726365751874520219978646\,9 \\ & 3899564749427740638459251925573263034537315482685\,0 \\ & 7917026122142913461670429214311602221240479274737\,7 \\ & 94080665351419597459856902143413 \end{aligned}$$

Exercise 4.24. In the context of RSA, suppose $n = 51983$. Find p and q, knowing that they are a pair of twin primes. (See Section 2.3 for definition.)

Exercise 4.25. Two companies are implementing RSA with $n_1 = 30227$ and $n_2 = 35657$, respectively. Suppose Cobra knows that they are sharing a common prime factor. How can he quickly factor both numbers?

Exercise 4.26. Still on RSA, suppose $n = 1520041$, given that p and q are quite close together. Find them using a factoring technique most suitable for this case.

4.4 RSA Cycling Attack [Project 4]

Alia announces her RSA public key, $n = 299$ and $e = 17$. When Bob sends her the message $s = 41^{17} \,\%\, 299 = 123$, Cobra intercepts it. Aware that factoring n is almost impossible, Cobra computes successive powering with the exponent e, starting with base s, as follows.

$$\begin{array}{crcl}
 & 123^{17} \,\%\, 299 = 197 & \rightarrow & 197^{17} \,\%\, 299 = 6 \\
\rightarrow & 6^{17} \,\%\, 299 = 288 & \rightarrow & 288^{17} \,\%\, 299 = 32 \\
\rightarrow & 32^{17} \,\%\, 299 = 210 & \rightarrow & 210^{17} \,\%\, 299 = 292 \\
\rightarrow & 292^{17} \,\%\, 299 = 119 & \rightarrow & 119^{17} \,\%\, 299 = 71 \\
\rightarrow & 71^{17} \,\%\, 299 = 41 & \rightarrow & 41^{17} \,\%\, 299 = 123 = s
\end{array}$$

From the last result, Cobra correctly concludes that $m = 41$.

Over the years, there have been various attempts to break the RSA cryptosystem. While none of these attacks is a serious blow to the system in general so far, a vast amount of research has also been done to study certain circumstances under which a specific implementation of the RSA becomes vulnerable. The above algorithm is one particular instance, which is given the name *cycling attack*. Fortunately, not for Cobra, this scheme is generally too slow to be effective.

Project 4.4.1. Repeat the above example with $n = 3161$ and $e = 3$. Use your own choice of m, and verify that Cobra does get it right again. Then prove that this algorithm always works and, in particular, that it returns the correct value of m.

Project 4.4.2. Attacks on the RSA cryptosystem is a subject of its own. Write a report paper on selected topics in this area. You may opt to include the cycling attack and provide some preventive ways to make the system immune to it.