Euler's totient and RSA

Exercises

Application of discrete models

Mathematical questions

1. What is a reduced residue system? Give *two* examples modulo 6.

2. Define $\varphi(n)$. Demonstrate the definition on $\varphi(6)$.

3. Calculate the following values of $\varphi()$:
a) $\varphi(25)$; b) $\varphi(27)$; c) $\varphi(30)$; d) $\varphi(36)$; e) $\varphi(45)$; f) $\varphi(81)$; g) $\varphi(100)$; h) $\varphi(600)$.

4. Write down Euler's totient theorem. Give *two* examples modulo 6.

5. Calculate the following modular powers using Euler's totient theorem:
a) $7^5 \bmod 6$; b) $137^{138} \bmod 10$; c) $52^{45} \bmod 25$; d) $321^{321} \bmod 100$; e) $1231^{642} \bmod 600$.

6. a) How does RSA generate the modulus (n)?
b) How does RSA generate the private exponent (d)?
c) How does RSA encrypt a message m?
d) How does RSA decrypt a cyphertext c?
e) Which of the following RSA parameters are public: d, e, n, p, q?

Programming exercises

7. Write a function that returns the list of invertible remainders for a given $m \in \mathbb{N}^+$ modulus. Print these lists from $m = 1$ to 20. What is the length of these lists?

8. a) Write a function that calculates $\varphi(n)$ using the definition.
b) Write a function that calculates $\varphi(n)$ using n's prime factorization.
c) Test both solutions from $n = 1$ to 100 using the built-in function `euler_phi()`.
d) $\varphi(n)$ has the property that it is even for all $n \geq 3$. Check this up to $n = 1000$.

9. What is $\sum_{d|n} \varphi(d)$, i.e. the sum of $\varphi(d)$ for all d positive divisors of n? Try to answer this by calculating the sum for many n values. Then check the guess for all n between 1 and 1000.

10. a) Generate an RSA key pair. For simplicity, you can use the (insecure) random number generator `random_prime(L)` to generate primes below L. First try it for $L = 1000$, and if it works, then finally change it to $L = 2^{120}$. Generate the public exponent by finding the first odd number that satisfies the criteria of RSA.
b) Encrypt a message m ($2 \leq m < n$) using the public key.
c) Decrypt the ciphertext c using the private key, i.e. restore the original m.

(Don't use any operation that would require breaking RSA, e.g. factoring n.)

11. Speed up the implementation of RSA using the chinese remainder theorem.

12. a) Write a function that checks whether a given $n \in \mathbb{N}^+$ is a prime number using the Fermat primality test. (Use a single random base a.)
b) Find an n for which the test gives an incorrect result.
c) Improve the test by repeating it k times (using separate random a's in each step), where k is also an input parameter.

13. Find all Fermat pseudoprimes to base $a = 2$ between 1 and 10000.

14. Find all Carmichael numbers between 1 and 10000.

15. Implement the Miller–Rabin primality test.

16. Find all strong pseudoprimes to base $a = 2$ between 1 and 10000.