Homework 4 for Modular Arithmetic: The RSA Cipher

Gregory V. Bard

April 25, 2018

- This is a practice workbook for the RSA cipher.
- It is not suitable for learning the RSA cipher from scratch. However, there are many websites that explain the RSA cipher.
- In the solutions of this packet, the black ink represents the question, and the blue ink represents the answer. The red ink is additional information that would normally only be calculated with technology (e.g. Sage), or extraordinary patience.
- Most problems have been cloned 3–4 times. A clone is an identical version of a problem with only the coefficients changed.
- All clones have the same number, but a different letter. So problems 1a, 1b, and 1c are clones of each other.
- Some problems cannot be cloned, because nothing changes.

Problem 1

Let's see how much you remember of the "alphabet soup" in RSA.

Problem 1a

Describe each of these with a short phrase.

- What is m?
- What is d?
- What is N?
- What is e?
- What is q?
- What is ϕ?
- What is c?
- What is p?

Problem 1b

For each of the following, tell me what is the standard choice of letter (from English or Greek) for representing that in RSA.

- The plaintext message.
- The encryption exponent.
- The decryption exponent.
- The two huge primes.
- The ciphertext message.
- The modulus.
- The Euler Totient Function of the modulus.

Problem 1

Let's see how much you remember of the "alphabet soup" in RSA.

Problem 1a

Describe each of these with a short phrase.

- What is m? The plaintext message.
- What is d? The decryption exponent.
- What is N? The modulus.
- What is e? The encryption exponent.
- What is q? One of the two huge primes.
- What is ϕ? The Euler Totient Function of the modulus.
- What is c? The ciphertext message.
- What is p? One of the two huge primes.

Problem 1b

For each of the following, tell me what is the standard choice of letter (from English or Greek) for representing that in RSA.

- The plaintext message. m
- The encryption exponent. e
- The decryption exponent. d
- The two huge primes. p, q
- The ciphertext message. c
- The modulus. N
- The Euler Totient Function of the modulus. ϕ

Problem 2a

Let $p = 101$ and $q = 103$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

Problem 2b

Let $p = 101$ and $q = 97$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

Problem 2c

Let $p = 103$ and $q = 107$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

Problem 2a

Let $p = 101$ and $q = 103$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

$$\begin{aligned} N &= (101)(103) = 10,403 \\ \phi &= (100)(102) = 10,200 \end{aligned}$$

For e, you can pick any integer coprime to ϕ. Accordingly, we have to factor ϕ, to learn what is safe.

$$10,200 = (100)(102) = (25)(4)(2)(51) = (5^2)(2^3)(3)(17)$$

As you can see, whatever e we choose must have no 2s, no 3s, no 5s, and no 17s in its prime factorization. The first few choices are

$$\{7, 11, 13, 19, 23, 29, 31, 37, 41, 43, \ldots\}$$

and you can just pick one, provided that $2 < e < 10,200$.

Whatever you choose for e, the formula for d should be e^{-1} mod ϕ. Note, that is mod ϕ, not mod N. Also, you should put your particular chosen number in place of the e.

Problem 2b

Let $p = 101$ and $q = 97$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

$$\begin{aligned} N &= (101)(97) = 9797 \\ \phi &= (100)(96) = 9600 \end{aligned}$$

For e, you can pick any integer coprime to ϕ. Accordingly, we have to factor ϕ, to learn what is safe.

$$9600 = (100)(96) = (25)(4)(4)(24) = (5^2)(4^3)(6) = (5^2)(2^7)(3)$$

As you can see, whatever e we choose must have no 2s, no 3s, and no 5s in its prime factorization. The first few choices are:

$$\{7, 11, 13, 17, 19, 23, 29, 31, 37, 41, \ldots\}$$

and you can just pick one, provided that $2 < e < 9600$.

Whatever you choose for e, the formula for d should be e^{-1} mod ϕ. Note, that is mod ϕ, not mod N. Also, you should put your particular chosen number in place of the e.

Problem 2c

Let $p = 103$ and $q = 107$. Compute a valid N, ϕ, e, and d. You can leave d as a formula, but I want numbers for the other three.

$$\begin{aligned} N &= (103)(107) = 11,021 \\ \phi &= (102)(106) = 10,812 \end{aligned}$$

For e, you can pick any integer coprime to ϕ. Accordingly, we have to factor ϕ, to learn what is safe.

$$10,812 = (102)(106) = (2)(51)(2)(53) = (2)(3)(17)(2)(53) = 2^2(3)(17)(53)$$

As you can see, whatever e we choose must have no 2s, no 3s, no 17s, and no 53s in its prime factorization. The first few choices are:

$$\{5, 7, 11, 13, 19, 23, 29, 31, 35, 37, \ldots\}$$

and you can just pick one, provided that $2 < e < 10,812$.

Whatever you choose for e, the formula for d should be $e^{-1} \bmod \phi$. Note, that is mod ϕ, not mod N. Also, you should put your particular chosen number in place of the e.

Problem 3

Now we'll learn whether or not you understand how the different outputs of the key-generation algorithm should be handled.

Problem 3a

A friend of yours is generating an RSA public-private key pair. He has followed all the steps of the key-generation process, but has forgotten what to do with each of the six things he has computed.

Of the six items computed during RSA key generation, p, q, N, ϕ, e, d, label each one as "public," "private," or "destroy."

- Circle one. For security, e should be: public private destroyed.
- Circle one. For security, d should be: public private destroyed.
- Circle one. For security, N should be: public private destroyed.
- Circle one. For security, Φ should be: public private destroyed.
- Circle one. For security, p should be: public private destroyed.
- Circle one. For security, q should be: public private destroyed.

Problem 3b

A friend of yours has completed the RSA key generation process, and has made a public-private key pair.

Of the six items computed during RSA key generation, p, q, N, ϕ, e, d, which should be made public? kept private? be destroyed?

Made Public: (list them)

Kept Private: (list them)

Destroyed: (list them)

Problem 3

Now we'll learn whether or not you understand how the different outputs of the key-generation algorithm should be handled.

Problem 3a

A friend of yours is generating an RSA public-private key pair. He has followed all the steps of the key-generation process, but has forgotten what to do with each of the six things he has computed.

Of the six items computed during RSA key generation, p, q, N, ϕ, e, d, label each one as "public," "private," or "destroy."

- Choose one. For security, e should be: public ✓ ~~private~~ ~~destroyed~~.
- Choose one. For security, d should be: ~~public~~ private ✓ ~~destroyed~~.
- Choose one. For security, N should be: public ✓ ~~private~~ ~~destroyed~~.
- Choose one. For security, Φ should be: ~~public~~ ~~private~~ destroyed ✓.
- Choose one. For security, p should be: ~~public~~ ~~private~~ destroyed ✓.
- Choose one. For security, q should be: ~~public~~ ~~private~~ destroyed ✓.

Warning: The six items can be presented in any order.

Problem 3b

A friend of yours has completed the RSA key generation process, and has made a public-private key pair.

Of the six items computed during RSA key generation, p, q, N, ϕ, e, d, which should be made public? kept private? be destroyed?

Made Public: (list them)

N, e

Kept Private: (list them)

d

Destroyed: (list them)

ϕ, p, q

Problem 4

Problem 4a:

1. Bob generates the following during key generation: $p = 103$, $q = 107$, $N = 11,021$, $\phi = 10,812$, but clearly he isn't finished yet. What are the requirements for e?
2. Okay, now Bob chooses $e = 2141$. What are the requirements for d?
3. Which of the six items will comprise Bob's public key? (Variable names are fine.)
4. Which of the six items will comprise Bob's private key? (Variable names are fine.)
5. As it comes to pass, Bob computes $d = 101$. Charlene sends him an encrypted message, $c = 9798$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)
6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 1234$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

Problem 4b:

1. Bob generates the following during key generation: $p = 101$, $q = 107$, $N = 10,807$, $\phi = 10,600$, but clearly he isn't finished yet. What are the requirements for e?
2. Okay, now Bob chooses $e = 2143$. What are the requirements for d?
3. Which of the six items will comprise Bob's public key? (Variable names are fine.)
4. Which of the six items will comprise Bob's private key? (Variable names are fine.)
5. As it comes to pass, Bob computes $d = 5807$. Charlene sends him an encrypted message, $c = 6076$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)
6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 5678$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

Problem 4c:

1. Bob generates the following during key generation: $p = 103$, $q = 107$, $N = 11,021$, $\phi = 10,812$, but clearly he isn't finished yet. What are the requirements for e?
2. Okay, now Bob chooses $e = 1009$. What are the requirements for d?
3. Which of the six items will comprise Bob's public key? (Variable names are fine.)
4. Which of the six items will comprise Bob's private key? (Variable names are fine.)
5. As it comes to pass, Bob computes $d = 8401$. Charlene sends him an encrypted message, $c = 6710$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)
6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 1234$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

Problem 4

Problem 4a:

1. Bob generates the following during key generation: $p = 103$, $q = 107$, $N = 11,021$, $\phi = 10,812$, but clearly he isn't finished yet. What are the requirements for e?

   We need e to be coprime to $\phi = 10,812$. This means that we have to factor 10,812 to understand the requirements further. We obtain

   $$10,812 = (102)(106) = (2)(51)(2)(53) = (2)(3)(17)(2)(53) = 2^2(3)(17)(53)$$

   which means that the factorization of e must have no 2s, no 3s, no 17s, and no 53s. Furthermore, $2 < e < 10,812$.

2. Okay, now Bob chooses $e = 2141$. What are the requirements for d?

   It is a simple calculation. We require $d = e^{-1} \bmod \phi$, which means $d = 2141^{-1} \bmod 10,812$. We need to compute the inverse of 2141 mod 10,812. Sage says that this is 101.

3. Which of the six items will comprise Bob's public key? (Variable names are fine.)

   Bob's public key will consist of N and e.

4. Which of the six items will comprise Bob's private key? (Variable names are fine.)

   Only d.

5. As it comes to pass, Bob computes $d = 101$. Charlene sends him an encrypted message, $c = 9798$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)

   Bob must compute $9798^{101} \bmod 11,021$. Sage says that this is 420.

6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 1234$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

   Alice must compute $1234^{2141} \bmod 11,021$. Sage says that this is 6643.

Problem 4b:

1. Bob generates the following during key generation: $p = 101$, $q = 107$, $N = 10,807$, $\phi = 10,600$, but clearly he isn't finished yet. What are the requirements for e?

   We need e to be coprime to $\phi = 10,600$. This means that we have to factor 10,600 to understand the requirements further. We obtain

   $$10,600 = (106)(100) = (2)(53)(4)(25) = (2^3)(5^2)(53)$$

   which means that the factorization of e must have no 2s, no 5s, and no 53s. Furthermore, $2 < e < 10,600$.

2. Okay, now Bob chooses $e = 2143$. What are the requirements for d?

   We require $d = e^{-1} \bmod \phi$, which means $d = 2143^{-1} \bmod 10,600$. We need to compute the inverse of 2143 mod 10,600. Sage says that this is 5807.

3. Which of the six items will comprise Bob's public key? (Variable names are fine.)

   Bob's public key will consist of N and e.

4. Which of the six items will comprise Bob's private key? (Variable names are fine.)

   Only d.

5. As it comes to pass, Bob computes $d = 5807$. Charlene sends him an encrypted message, $c = 6076$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)

   Bob must compute $6076^{5807} \bmod 10{,}807$. Sage says that this is 888. This means "good fortune" or "lots of money" in some classical Chinese numerology.

6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 5678$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

   Alice must compute $5678^{2143} \bmod 10{,}807$. Sage says that this is 6973.

Problem 4c:

1. Bob generates the following during key generation: $p = 103$, $q = 107$, $N = 11,021$, $\phi = 10,812$, but clearly he isn't finished yet. What are the requirements for e?

   We need e to be coprime to $\phi = 10,812$. This means that we have to factor 10,812 to understand the requirements further. We obtain

   $$10,812 = (102)(106) = (2)(51)(2)(53) = (2^2)(3)(17)(53)$$

   which means that the factorization of e must have no 2s, no 3s, no 17s, and no 53s. Furthermore, $2 < e < 10,812$.

2. Okay, now Bob chooses $e = 1009$. What are the requirements for d?

   We require $d = e^{-1} \bmod \phi$, which means $d = 1009^{-1} \bmod 10,812$. We need to compute the inverse of 1009 mod 10,812. Sage says that this is 8401.

3. Which of the six items will comprise Bob's public key? (Variable names are fine.)

   Bob's public key will consist of N and e.

4. Which of the six items will comprise Bob's private key? (Variable names are fine.)

   Only d.

5. As it comes to pass, Bob computes $d = 8401$. Charlene sends him an encrypted message, $c = 6710$. What does Bob have to compute to decipher the message? (Your answer should be a formula, with numbers but not variables.)

   Bob must compute $6710^{8401} \bmod 11{,}021$. Sage says that this is 4321.

6. Alice wants to send Bob a secret message. She downloads his public key. The message is $m = 1234$. What should she compute in order to encrypt this message for him? (Your answer should be a formula, with numbers but not variables.)

   Alice must compute $1234^{1009} \bmod 11{,}021$. Sage says that this is 146. It is a small number. Indeed, this does sometimes happen.

Problem 5

There is a distrustful user, Vincent. He doesn't understand why it is so important that ϕ, p, and q get destroyed.

5.1: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's ϕ becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

5.2: Okay, so now we understand that ϕ should be destroyed. We learn that RSA is very dependent on the hardness of factoring. Why would it be so bad if factoring huge numbers became easy?! In particular, what would happen if nothing other than Vincent's public key was public, but factoring huge numbers was easy?

5.3: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's q becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

5.4: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's p becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

Problem 5

There is a distrustful user, Vincent. He doesn't understand why it is so important that ϕ, p, and q get destroyed.

5.1: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's ϕ becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

We know that e is part of the public key. If ϕ becomes public also, then one can compute $e^{-1} \bmod \phi$. This is the inverse of $e \bmod \phi$. Clearly, the number obtained would be d. Whoever does this computation now has Vincent's d, which means that they can read his messages and probably sign documents, such as checks, as if they were him.

5.2: Okay, so now we understand that ϕ should be destroyed. We learn that RSA is very dependent on the hardness of factoring. Why would it be so bad if factoring huge numbers became easy?! In particular, what would happen if nothing other than Vincent's public key was public, but factoring huge numbers was easy?

Again, we know that e and N comprise the public key. If factoring became easy, someone could compute p and q by factoring $N = pq$, because N is public after all. Then since the attacker knows p and q, it is extremely easy to compute $\phi = (p-1)(q-1)$. Once the attacker knows ϕ, they can do what we wrote in the answer to Subproblem 3.1.

5.3: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's q becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

Given q, simple division would reveal p. Because $N = pq$, the attacker just needs to compute $N/q = p$. Recall, N is public after all. Also, note that this is ordinary division, in the sense of the ordinary integers, not modular arithmetic. Then since the attacker knows p and q, it is extremely easy to compute $\phi = (p-1)(q-1)$. Once the attacker knows ϕ, they can do what we wrote in the answer to Subproblem 3.1.

5.4: Let's assume that no one publishes an algorithm to make it easier to factor huge numbers. Suppose Vincent's p becomes public, but nothing else (besides Vincent's public key) becomes public. What could happen then?

This is clearly the same question as Subproblem 3.3. Given p, simple division would reveal q. Because $N = pq$, the attacker just needs to compute $N/p = q$. Then since the attacker knows p and q, it is extremely easy to compute $\phi = (p-1)(q-1)$. Once the attacker knows ϕ, they can do what we wrote in the answer to Subproblem 3.1.

Moral: What is the moral of the story? Do not violate the requirements of the algorithm under any circumstances!

Question 6

Whenever you use your web browser in `https` mode, and see that key icon or lock icon that indicates an encrypted communication, then you're using either RSA, or a related algorithm called Diffie-Hellman. The mechanism for doing this is a protocol called SSL (the secure socket's layer) and its successor, TLS (transport level security).

In 2012, some researchers (see bibliographic references below) took a bunch of RSA public keys (11,400,000 of them) from the Electronic Frontier Foundation's SSL Observatory, and started taking gcds of pairs of Ns. It turns out that this exceptionally simple operation—downloading lots of public keys, picking random pairs of Ns, and taking gcds—broke 0.2% of public keys. That's a lot larger than it sounds like. There were 26,965 that were broken, which is alarming since most programmers who use cryptography think that RSA is unbreakable.

Question 6a

Suppose $e_1 = 151,153$ and $e_2 = 176,303$, but $N_1 = 302,303$ and $N_2 = 352,603$. You compute $\gcd(N_1, N_2) = 503$ using the Extended Euclidean Algorithm, or Sage.

- What is the factorization of N_1?
- What is ϕ_1?
- How can we obtain the private key, d_1?
- What is the factorization of N_2?
- What is ϕ_2?
- How can we obtain the private key, d_2?

Question 6b

Suppose $e_1 = 386,329$ and $e_2 = 387,197$, but $N_1 = 772,637$ and $N_2 = 774,391$. You compute $\gcd(N_1, N_2) = 877$ using the Extended Euclidean Algorithm, or Sage.

- What is the factorization of N_1?
- What is ϕ_1?
- How can we obtain the private key, d_1?
- What is the factorization of N_2?
- What is ϕ_2?
- How can we obtain the private key, d_2?

Citations

A. Lenstra, J. Hughes, M. Augier, J. Bos, T. Kleinjung, and C. Wachte. "Ron was wrong, Whit is right." Posted to `eprint.iacr.org`, 17 February, 2012.

`https://eprint.iacr.org/2012/064.pdf`

There is also a NY Times article about this: J. Markoff. "Flaw Found in an Online Encryption Method." *The New York Times*. February 15th, 2012.

`http://www.nytimes.com/2012/02/15/technology/researchers-find-flaw-in-an-online-encryption-method.html`

Question 6

See the previous page for the background on Question 6.

Question 6a

Suppose $e_1 = 151,153$ and $e_2 = 176,303$, but $N_1 = 302,303$ and $N_2 = 352,603$. You compute $\gcd(N_1, N_2) = 503$ using the Extended Euclidean Algorithm, or Sage.

- What is the factorization of N_1? Because $N_1/g = 302,303/503 = 601$, we know $(503)(601) = 302,303$.
- What is ϕ_1? We compute $(502)(600) = 301,200$.
- How can we obtain the private key, d_1? Compute the inverse of 151,153 mod 301,200, namely $151,153^{-1}$ mod 301,200. According to Sage, $d = 817$.
- What is the factorization of N_2? Because $N_2/g = 352,603/503 = 701$, we know $(503)(701) = 352,603$.
- What is ϕ_2? We compute $(502)(700) = 351,400$.
- How can we obtain the private key, d_2? Compute the inverse of 176,303 mod 351,400, namely $176,303^{-1}$ mod 351,400. According to Sage, $d = 24,767$.

Note: Now you have both private keys, and can read their mail and sign checks as if you were them!

Question 6b

Suppose $e_1 = 386,329$ and $e_2 = 387,197$, but $N_1 = 772,637$ and $N_2 = 774,391$. You compute $\gcd(N_1, N_2) = 877$ using the Extended Euclidean Algorithm, or Sage.

- What is the factorization of N_1? Because $N_1/g = 772,637/877 = 881$, we know $(877)(881) = 772,637$.
- What is ϕ_1? We compute $(876)(880) = 770,880$.
- How can we obtain the private key, d_1? Compute the inverse of 386,329 mod 770,880, namely $386,329^{-1}$ mod 770,880. According to Sage, $d = 352,489$.
- What is the factorization of N_2? Because $N_2/g = 774,391/877 = 883$, we know $(877)(883) = 774,391$.
- What is ϕ_2? We compute $(876)(882) = 772,632$.
- How can we obtain the private key, d_2? Compute the inverse of 387,197 mod 772,632, namely $387,197^{-1}$ mod 772,632. According to Sage, $d = 231,965$.

Note: Now you have both private keys, and can read their mail and sign checks as if you were them!

Citations

A. Lenstra, J. Hughes, M. Augier, J. Bos, T. Kleinjung, and C. Wachte. "Ron was wrong, Whit is right." Posted to `eprint.iacr.org`, 17 February, 2012.
`https://eprint.iacr.org/2012/064.pdf`
There is also a NY Times article about this: J. Markoff. "Flaw Found in an Online Encryption Method." *The New York Times*. February 15th, 2012.
`http://www.nytimes.com/2012/02/15/technology/researchers-find-flaw-in-an-online-encryption-method.html`

Question 7

We return to Vincent, who doesn't trust all the details of the algorithm specification. When we require that $2 < e < \phi$, we are explicitly ruling out

$$e \in \{0, 1, 2, \phi\}$$

7.1: What would go wrong if $e = 1$?

7.2: What would go wrong if $e = 0$?

7.3: What would go wrong if $e = \phi$?

7.4: What would go wrong if $e = 2$?

Question 7

We return to Vincent, who doesn't trust all the details of the algorithm specification. When we require that $2 < e < \phi$, we are explicitly ruling out

$$e \in \{0, 1, 2, \phi\}$$

7.1: What would go wrong if $e = 1$?

When you encrypt a message, the ciphertext is $c = m^e \bmod N$. We would have $c = m^1 = m \bmod N$. Your messages would not be encrypted—you would be transmitting your messages in plaintext.

7.2: What would go wrong if $e = 0$?

When you encrypt a message, the ciphertext is $c = m^0 \bmod N$. We would have $c = m^0 = 1 \bmod N$. The ciphertext would be 1 regardless of the message. The receiver would have no idea which message you wanted to send, because all messages turn into a 1. This is like the sender putting a letter into an envelope, and burning it, rather than mailing it. While no one will be able to read the original message, the intended receiver will have no idea what the message contains.

7.3: What would go wrong if $e = \phi$?

When you encrypt a message, the ciphertext is $c = m^\phi \bmod N$. We would have $c = m^\phi = m^0 = 1 \bmod N$. That's because exponents, when working mod N, should be reduced mod ϕ, and $\phi = 0 \bmod \phi$. The ciphertext would be 1 regardless of the message, like the previous subproblem.

7.4: What would go wrong if $e = 2$?

Since p and q are huge primes, they are odd. This means that $(p-1)$ and $(q-1)$ are even. The product of two even numbers is even. Since $\phi = (p-1)(q-1)$ is even, and $e = 2$ is even, then $\gcd(e, \phi) = \gcd(2, \phi) = 2$. This means $\gcd(e, \phi) \neq 1$ and therefore, d will not exist. The receiver cannot decode the message, because d does not exist.