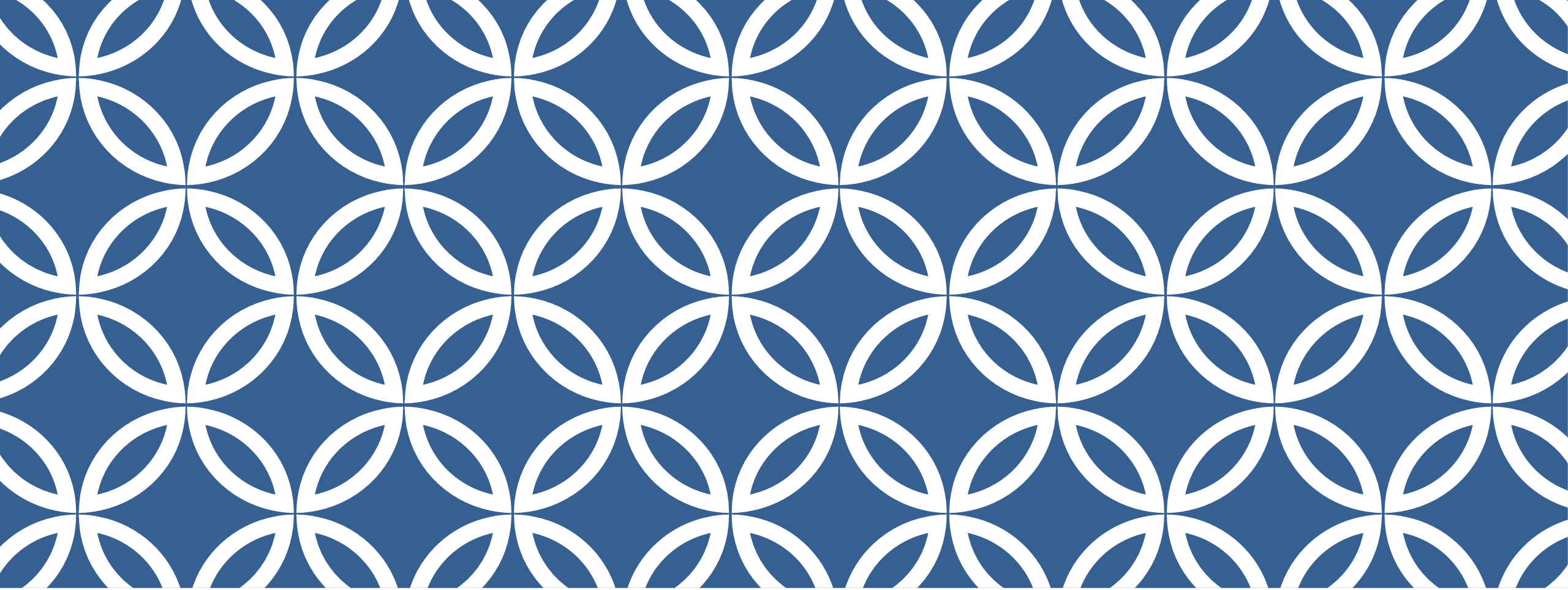


RSA ALGORITHM TO ENCRYPT AND DECRYPT THE DATA

OBJECTIVE

To implement RSA Algorithm to Encrypt and Decrypt data.

INTRODUCTION

- **Internet Security is becoming more important everyday.**
- **Cryptography is the heart of security.**
- **In order to protect our data, encryption is done at sender side and decryption is carried out at receiver's side.**

INTRODUCTION

- **SECURITY ASPECTS.**
  - **AUTHENTICATION.**
    - Authentication provides the identification of the originator.
    - It confirms to the receiver that the data received has been sent only by an identified and verified sender.
  - **CONFIDENTIALITY.**
    - Confidentiality is the fundamental security service provided by cryptography.
    - It is a security service that keeps the information confidential from an unauthorized person.
    - It is sometimes referred to as ***privacy*** or ***secrecy.***
  - **NONREPUDIATION.**
    - It is a security service that ensures that an entity cannot refuse the ownership of a previous commitment or an action.
    - It is an assurance that the original creator of the data cannot deny the creation or transmission of the said data to a recipient or third party.
  - **DATA INTEGRITY.**
    - It is security service that deals with identifying any alteration to the data.
    - The data may get modified by an unauthorized entity intentionally or accidently.
    - Integrity service confirms that whether data is intact or not since it was last created, transmitted, or stored by an authorized user.

INTRODUCTION

- **HASHING.**
  - **Creating a miniature version of a message instead of original message.**
- **AUTHENTICATION OF PEOPLE.**
- **KEY MANAGEMENT**

INTRODUCTION

- **SECURITY AND INTERNET MODEL.**
  - **OSI model provided encryption/ decryption services in the presentation layer which does not exists in the internet model.**
  - **At which layer security need to be implemented in internet model?**
  - **Security breach can happen in any of the five layers.**
  - **At physical layer, an intruder can wiretap into the transmission media and read or alter a sequence of bits.**
  - **At data link layer, frames can be captured and read or altered.**
    - **Example: LAN, where transmission is broadcast and every station receives a copy of frame.**
  - **At the network layer, an IP datagram can be removed, altered, or inserted into the network.**
  - **At the transport layer, a user datagram or a segment can be captured or altered.**
  - **Finally, at application layer, the whole message can be altered or read.**

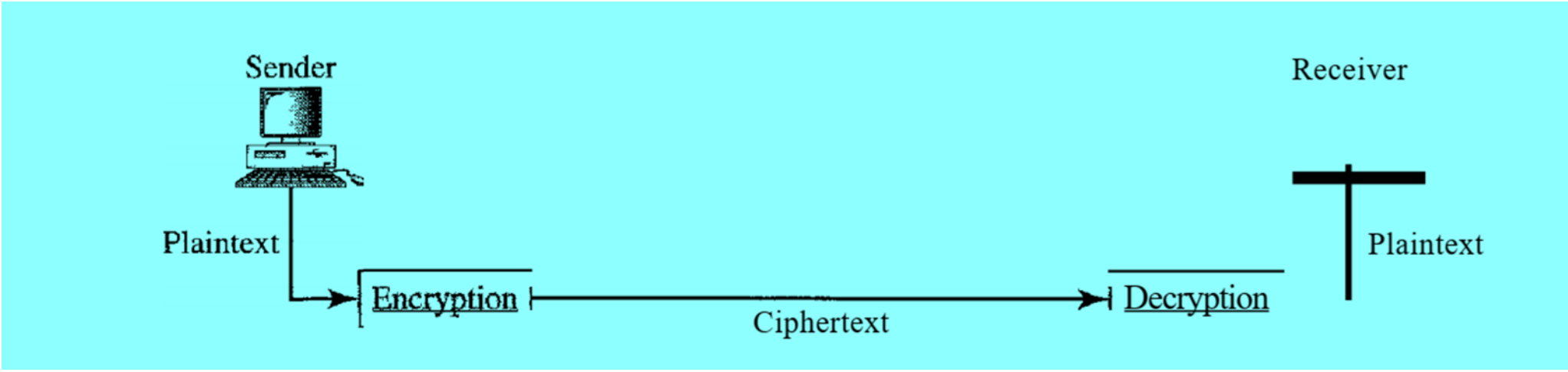
INTRODUCTION

- Concerns need be addressed before adding security to the Internet.
  - Security is more effective in which layer?
  - Security is easier to provide in which layer?
  - Security at one level, is that enough?

CRYPTOGRAPHY

- Cryptography in Greek means "Secret Writing".
- It is an art and science of transforming messages to make them secure and immune to attacks.

CRYPTOGRAPHY COMPONENTS BLOCK DIAGRAM



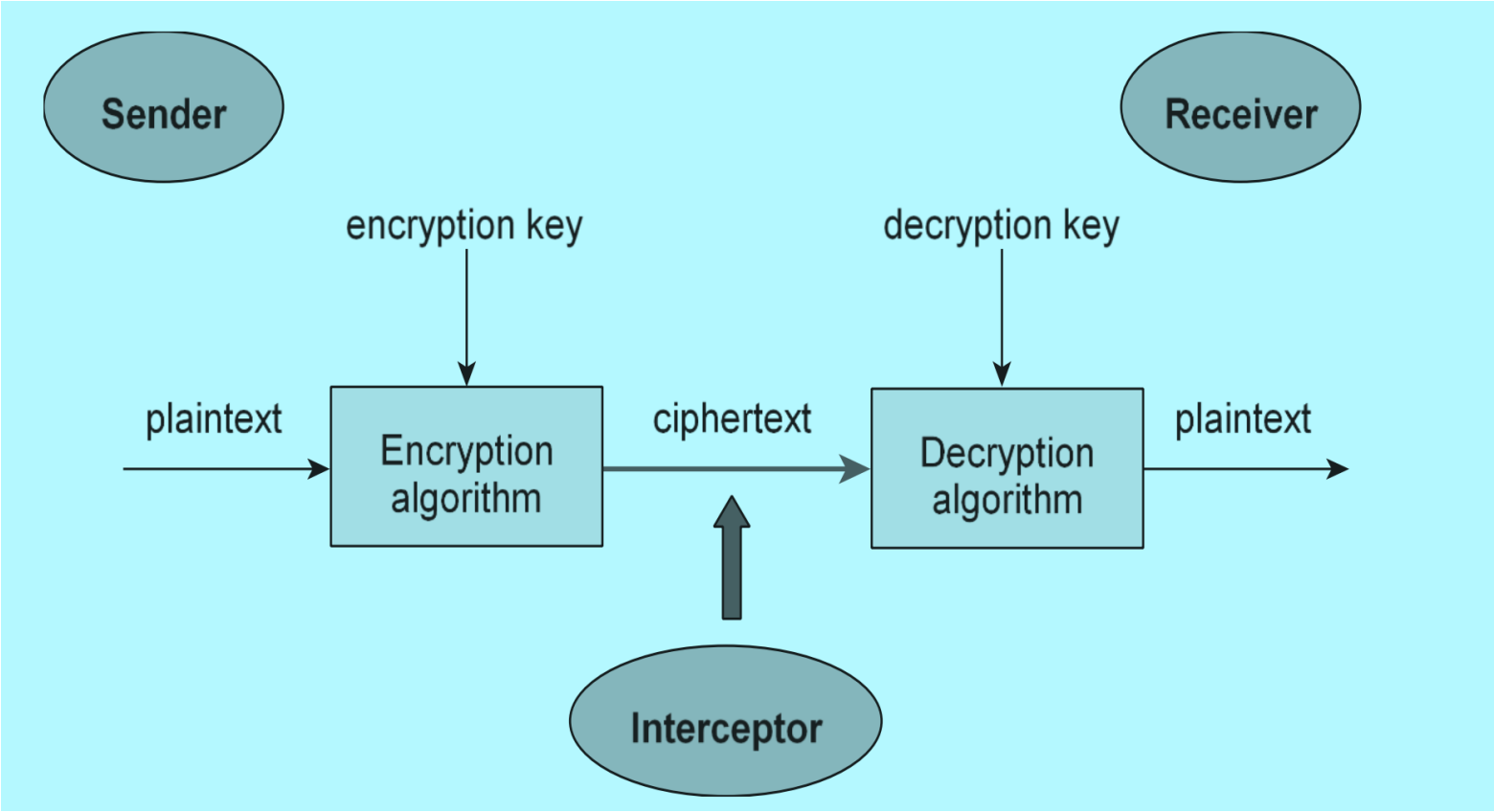
CRYPTOGRAPHY COMPONENTS

- The original message, before being transformed, is called ***plaintext.***
- After the message is transformed, it is called ***ciphertext.***
- An ***encryption algorithm*** transforms the plaintext to ciphertext;
- A ***decryption algorithm*** transforms the ciphertext back to plaintext.
- The sender uses an encryption algorithm and the receiver uses a decryption algorithm.
- The term ***cipher*** refers to encryption/ decryption algorithms.
- A ***key*** is a number(value) that the cipher, as an algorithm, operates on.

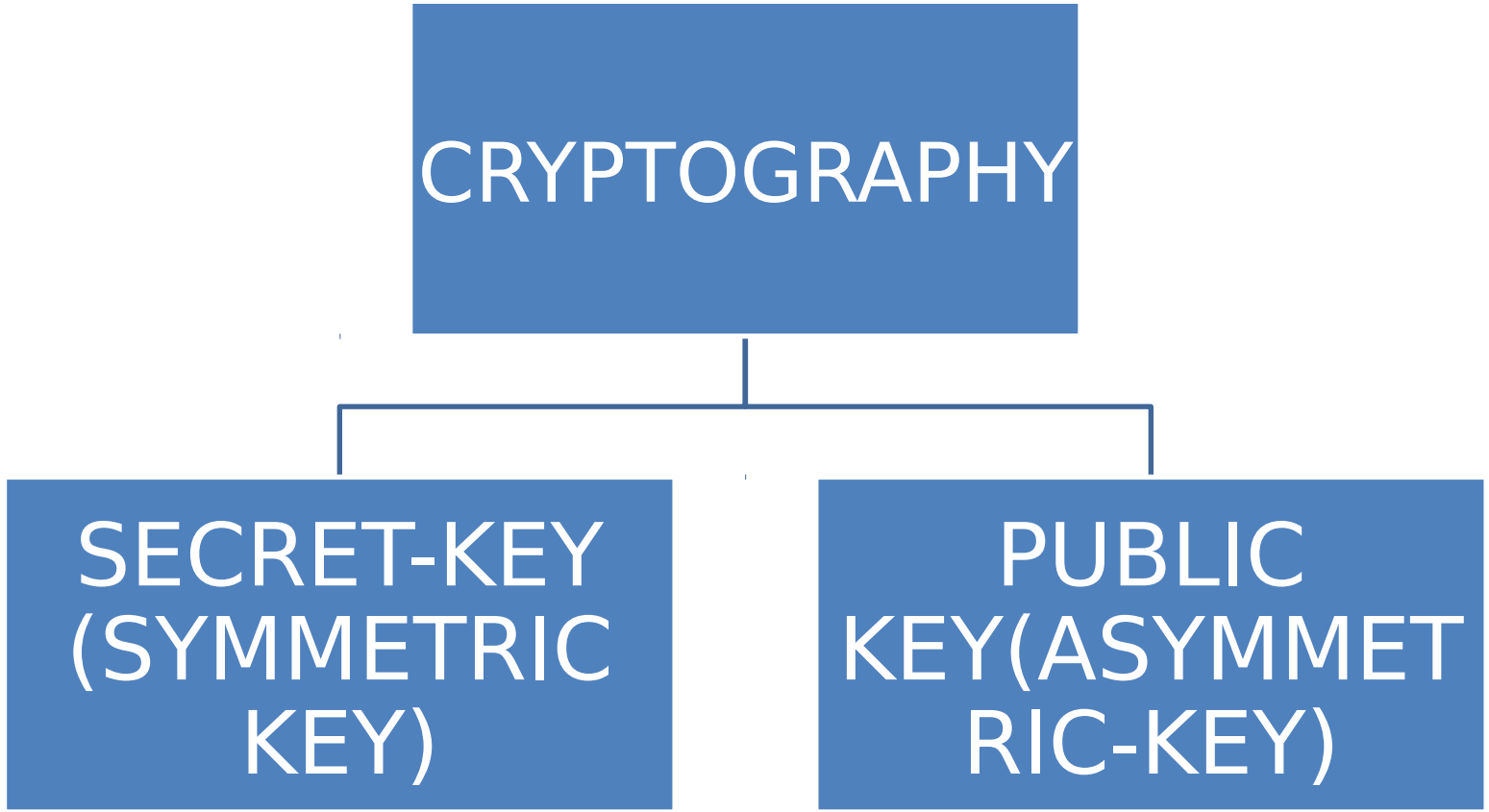
ENCRYPTION AND DECRYPTION

- To encrypt a message, *an encryption algorithm*, *encryption key* and *plaintext* is required.
- To decrypt a message, a *decryption algorithm*, a *decryption key* and *ciphertext* is required.
- The encryption and decryption algorithms are public; anyone can access them.
- The keys are secret; they need to be protected.

ENCRYPTION AND DECRYPTION BLOCK DIAGRAM



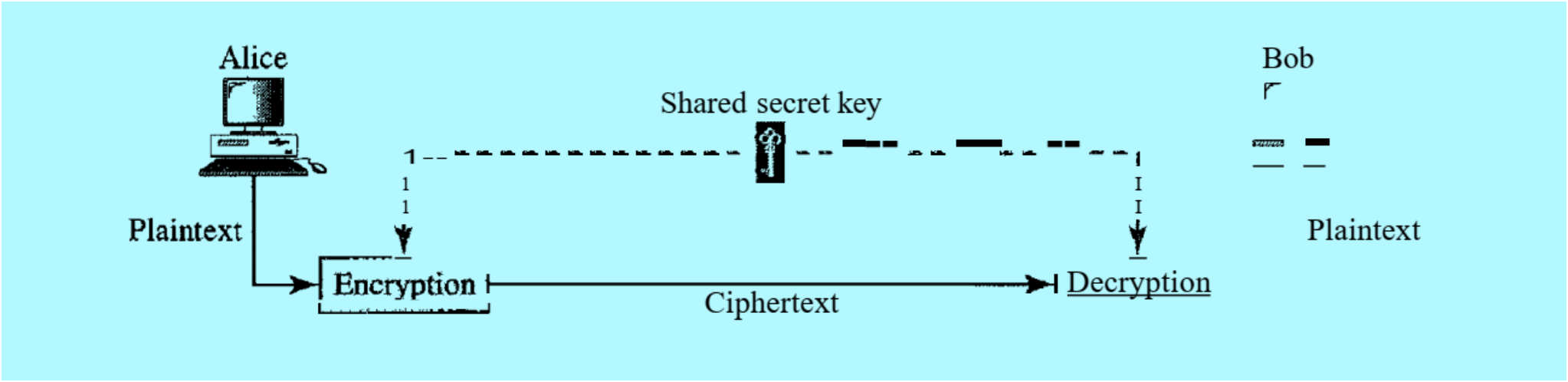
TYPES OF CRYPTOGRAPHY



SYMMETRIC KEY(SECRET-KEY) CRYPTOGRAPHY

- In symmetric key cryptography, the same key is used by both sender and receiver.
- Sender uses this key and an encryption algorithm to encrypt data;
- Receiver uses the same key and a decryption algorithm to decrypt data.
- The key is shared.

SYMMETRIC KEY CRYPTOGRAPHY BLOCK DIAGRAM



DIFFERENCES BETWEEN SYMMETRIC-KEY AND ASYMMETRIC-KEY CRYPTOGRAPHY

Symmetric-key

- Same key is used for encryption and decryption.
- Used for long messages.
- Each pair of users must have a unique symmetric key: for ‘N’ users there need to be [N(N-1)/2] symmetric keys.
- Key distribution between sender and receiver is difficult.
- Simple algorithms.
- Ex: DES, Triple DES.

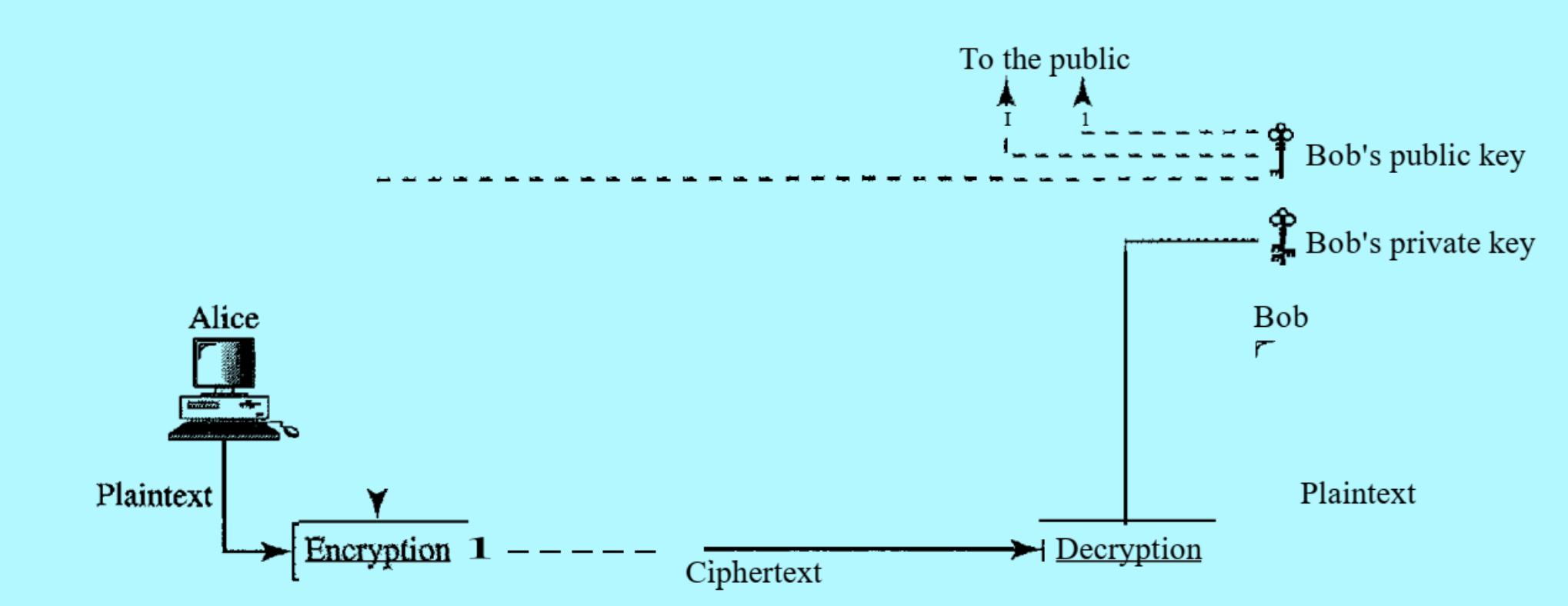
Asymmetric-key

- Two different keys are used for encryption and decryption: Public key and Private key.
- More efficient for short messages.
- Private key is kept by receiver, public key is announced to the public.
- Overcomes the drawback of symmetric-key cryptography, by reducing the number of keys to 2N.
- Complex algorithms.
- Ex: RSA algorithm.

ASYMMETRIC KEY (PUBLIC-KEY)CRYPTOGRAPHY

- In public-key cryptography, there are two key: a private key and a public key.
- The public key is used for encryption and private key for decryption.
- Advantages:
  - Removes the restriction of a shared symmetric key between two entities who need to communicate with each other.
  - Each entity creates a pair of keys: the private one is kept, and public one is distributed.
  - Each entity is independent, and a pair of keys created can be used to communicate with any other entity.
  - The number of keys needed is reduced tremendously to ‘2N’ only for ‘N’ users.
- Disadvantages:
  - Complexity of algorithm: needs large numbers.
  - Calculating ciphertext from plaintext using the long keys takes lot of time. Hence not recommended for large amounts of text.
  - The association between an entity and its public key must be verified. (can be overcome using certification authority)

ASYMMETRIC KEY (PUBLIC-KEY)CRYPTOGRAPHY BLOCK DIAGRAM



RSA ALGORITHM (Rivest, Shamir and Adleman)- EXAMPLE: GENERATING PUBLIC KEY

1. Choose two distinct prime numbers, **p**, and **q**.
2. Compute **n = p*q .**
3. Compute the totient of the product as **λ(n) = (p − 1) * (q − 1).**
   λ(n) can also be, λ(n) = lcm(λ(p), λ(q)) = lcm(p − 1, q − 1).
4. Choose any number 1 < e < λ(n) that is coprime to λ(n), e.

Coprime: two integers a and b are said to be relatively, mutually or co prime if the only positive integer (factor) that divides both of them is 1 a, and b themself need not be prime.

Example 14, 15 are co prime, but they themself are not prime, only common divisor is 1.

(n , e) is the Public key

RSA ALGORITHM- EXAMPLE: GENERATING PRIVATE KEY

5. Compute d, the modular multiplicative inverse of e (mod λ(n))

Inverse
Multiplicative
Modular
d * e mod λ(n) = 1

(n , d) is the Private key.

RSA ALGORITHM- EXAMPLE: ENCRYPTION/ DECRYPTION

- To encrypt, message m, into cipher c, c = (m power e) mod n
  using public key (n, e) at sender
- To decrypt, cipher text c to message m, m = (c power d) mod n
  using private key, (n, d) at receiver

RSA ALGORITHM- EXAMPLE:

p = 3, q = 11, n = 33
λ(n) = (p − 1) * (q − 1) = (3-1)*(11-1) = 2*10 = 20
Choose any number 1 < e < λ(n) that is coprime to λ(n), e = 3
(n , e), is the Public key = (33, 3)
Compute d, the modular multiplicative inverse of e (mod λ(n))
d such that, d * e mod λ(n) = 1
d * e mod λ(n) = d * 3 mod 20 = 7 * 3 mod 20 = 21 mod 20 = 1
d = 7
(n , d), is the Private key = (33, 7)

RSA ALGORITHM-EXAMPLE:

To encrypt, message m to Cipher text c, c = (m power e) mod n

Consider m = 2 3 4 , say its b c d, b is 2, c is 3 and d is 4
c = (m power 3) mod 33

c = (2 power 3) mod 33 = 8
c = (3 power 3) mod 33 = 27
c = (4 power 3) mod 33 = 64 mod 33 = 31

For message m = 2 3 4 , Cipher c = 8 27 31

RSA ALGORITHM-EXAMPLE:

To decrypt, cipher text c back to message m, m = (c power d) mod n

m = (c power 7) mod 33

c = 8 27 31

m = (8 power 7) mod 33 = 2
m = (27 power 7) mod 33 = 3
m = (31 power 7) mod 33 = 4

For Cipher c = 8 27 31 , message m = 2 3 4

SOURCE CODE

```
int main()
 {
   int p, q, n, lambdaN, d, e, length, i;
   int message[10], cipher[10];

   printf("\n Enter two distinct prime numbers p and q: ");
   scanf("%d%d", &p, &q); // Choose two distinct prime numbers, p, and q

   n = p*q;                    // Compute n = p*q
   lambdaN = (p - 1) * (q - 1); // Compute totient of the product λ(n)=(p−1)*(q−1)

   printf("\n Enter the Public and Private key, e and d, such that e and ");
   printf("(p-1)*(q-1) are co prime, and d * e mod (p-1)*(q-1) = 1 : ");
   scanf("%d%d", &e, &d);

   printf("\n Enter length of message: ");     scanf("%d", &length);

   printf("\n Enter the message, input 1 for a, 2 for b . . , ");
   printf("separated by space or line: ");
   for( i = 0; i < length; i++ )          scanf("%d", &message[i]);

   printf("\n At sender, encrypt message to cipher, cipher = ");
   for( i = 0; i < length; i++ )
     {
       cipher[i] = ( (long int) pow( message[i], e ) ) % n;
       printf("%d ", cipher[i]);
     }

   printf("\n At receiver, decrypt cipher to message, message = ");
   for( i = 0; i < length; i++ )
       printf("%ld ", ( (long int) pow( cipher[i], d ) ) % n );

   return 0;
 }
```

```c
#include <stdio.h>
#include <math.h>

int main()
 {
   int p, q, n, lambdaN, d, e, length, i;
   int message[10], cipher[10];

   printf("\n Enter two distinct prime numbers p and q: ");
   scanf("%d%d", &p, &q); // Choose two distinct prime numbers, p, and q

   n = p*q;                  // Compute n = p*q
   lambdaN = (p - 1) * (q - 1); // Compute totient of the product λ(n)=(p−1)*(q−1)

   printf("\n Enter the Public and Private key, e and d, such that e and ");
   printf("(p-1)*(q-1) are co prime, and d * e mod (p-1)*(q-1) = 1 : ");
   scanf("%d%d", &e, &d);

   printf("\n Enter length of message: ");    scanf("%d", &length);
```

```
  printf("\n Enter the message, input 1 for a, 2 for b . . , ");
  printf("separated by space or line: ");
  for( i = 0; i < length; i++ )          scanf("%d", &message[i]);

  printf("\n At sender, encrypt message to cipher, cipher = ");
  for( i = 0; i < length; i++ )
    {
      cipher[i] = ( (long int) pow( message[i], e ) ) % n;
      printf("%d ", cipher[i]);
    }

  printf("\n At receiver, decrypt cipher to message, message = ");
  for( i = 0; i < length; i++ )
      printf("%ld ", ( (long int) pow( cipher[i], d ) ) % n );

  return 0;
}
```

```c
#include <stdio.h>
#include <math.h>
#include <string.h>

int gcd(int m , int n )// ALGORITHM Euclid(m, n), gcd: greatest common divisor
 {                     //   Computes gcd(m, n) by Euclid's algorithm
   int r = 0;          //   Input: Two nonnegative, not-both-zero integers m and n
   char temp;          //   Output: Greatest common divisor of m and n
   while( n!=0 )       //   while n != 0
    {                  //     do
      r = m % n;       //       r ← m mod n
      m = n;           //       m ← n
      n = r;           //       n ← r
    }                  //     done
   return m;           //   return m
 }
```

```c
int main()
 {
   int p, q, n, lambdaN, d, e, length, i;
   char string[20];
   int message[20], cipher[20];

   printf("\n Enter two distinct prime numbers p and q: ");
   scanf("%d%d",&p, &q); // Choose two distinct prime numbers, p, and q

   n = p*q;                  // Compute n = p*q
   lambdaN = (p - 1) * (q - 1); // Compute totient of the product λ(n)=(p-1)*(q-1)

   // choose number e, such that 1 < e < λ(n) and is coprime to λ(n)
   // Find e, such that gcd( e, λ(n) ) = 1, Greatest common divisor of e and λ(n)
   // Two numbers e and λ(n) are co prime if gcd( e, λ(n) ) = 1,
   // That is no common divisor other than 1

   e = 2;   // e, such that, 1 < e < λ(n) and coprime to λ(n)
   while ( gcd( e, lambdaN ) != 1 && e < lambdaN )
     {
       e++;
     }
   printf("\n Public key = ( %d, %d )", n, e);
```

```
d = 1;  // Private key, d, such that,    d * e mod λ(n) = 1
while ( ( ( d * e ) % lambdaN ) != 1 )
  {
    d++;
  }
printf("\n Private key = ( %d, %d )", n, d);

printf("\n Enter the message, lower case characters, no space in between: ");
scanf("%s", string );

length = strlen(string);

for( i = 0; i < length; i++ )
  {
    message[i] = string[i] - 'a' ; // save a as 1, b as 2, c as 3 ...
  }

printf("\n At sender, encrypt message to cipher, cipher = ");
for( i = 0; i < length; i++ )
  {
    cipher[i] = ( (long int) pow( message[i], e ) ) % n;
    printf("\n %c as %d ", message[i] + 'a', cipher[i] );
  }
```

```
  printf("\n At receiver, decrypt cipher to message, message = ");
  for( i = 0; i < length; i++ )
    {
      printf("\n %d as %c ", cipher[i],
                      (char)( ( (long int) pow( cipher[i], d ) % n ) + 'a' ) );
    }

  return 0;
}

        /*Output:   ./a.out
          Enter two distinct prime numbers p and q: 3 11
          Public key = ( 33, 3 )
          Private key = ( 33, 7 )
          Enter the message, lower case characters, no space in between: dvg

          At sender, encrypt message to cipher, cipher =
          d as 27
          v as 21
          g as 18

          At receiver, decrypt cipher to message, message =
          27 as d
          21 as v
          18 as g
        */
```

EXPECTED OUTPUT CASE:1

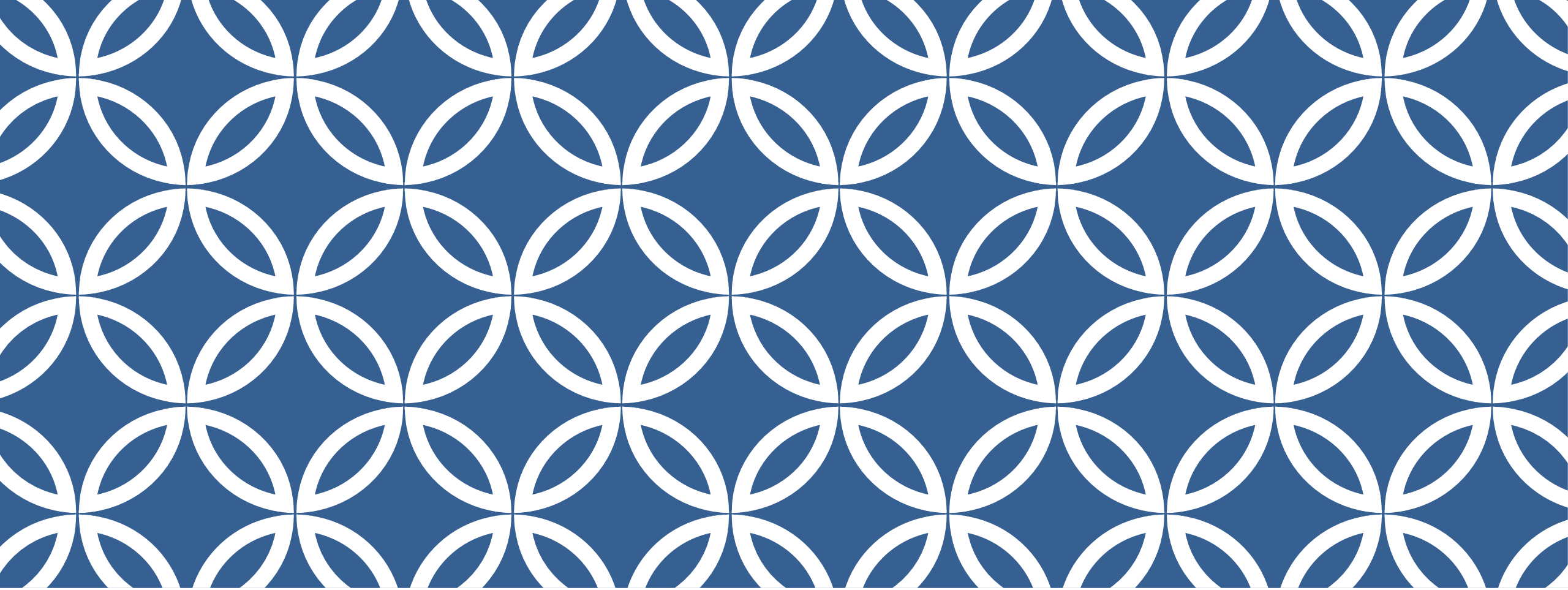
ENTER TWO DISTINCT PRIME NUMBERS P AND Q: **13 17**
N=221;
ENTER THE PUBLIC AND PRIVATE KEY, E AND D, SUCH THAT E AND (P-1)*(Q-1) ARE CO PRIME, AND D * E MOD (P-1)*(Q-1) = **1: 35**
11
ENTER LENGTH OF MESSAGE: **3**
ENTER THE MESSAGE, INPUT 1 FOR A, 2 FOR B . . , SEPARATED BY SPACE OR LINE: **1 2 3**
AT SENDER, ENCRYPT MESSAGE TO CIPHER, CIPHER = **1 59 61**
AT RECEIVER, DECRYPT CIPHER TO MESSAGE, MESSAGE = **1 2 3**

EXPECTED OUTPUT CASE:2

ENTER TWO DISTINCT PRIME NUMBERS P AND Q: **3 11**
N=33;
ENTER THE PUBLIC AND PRIVATE KEY, E AND D, SUCH THAT E AND (P-1)*(Q-1) ARE CO PRIME, AND D * E MOD (P-1)*(Q-1) = **1: 3 7**
ENTER LENGTH OF MESSAGE: **3**
ENTER THE MESSAGE, INPUT 1 FOR A, 2 FOR B . . , SEPARATED BY SPACE OR LINE:**2 3 4**
AT SENDER, ENCRYPT MESSAGE TO CIPHER, CIPHER =**8 27 31**
AT RECEIVER, DECRYPT CIPHER TO MESSAGE, MESSAGE = **2 3 4**

REFERENCES

1. Behrouz Forouzon - Data Communications and Networking, McGraw Hill Edition
2. Anany Levitin, Introduction to the design & analysis of algorithms
3. Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest, Clifford Stein Introduction to Algorithms



EXPECTED OUTCOME

Students will be able to implement RSA Algorithm to Encrypt and Decrypt data.