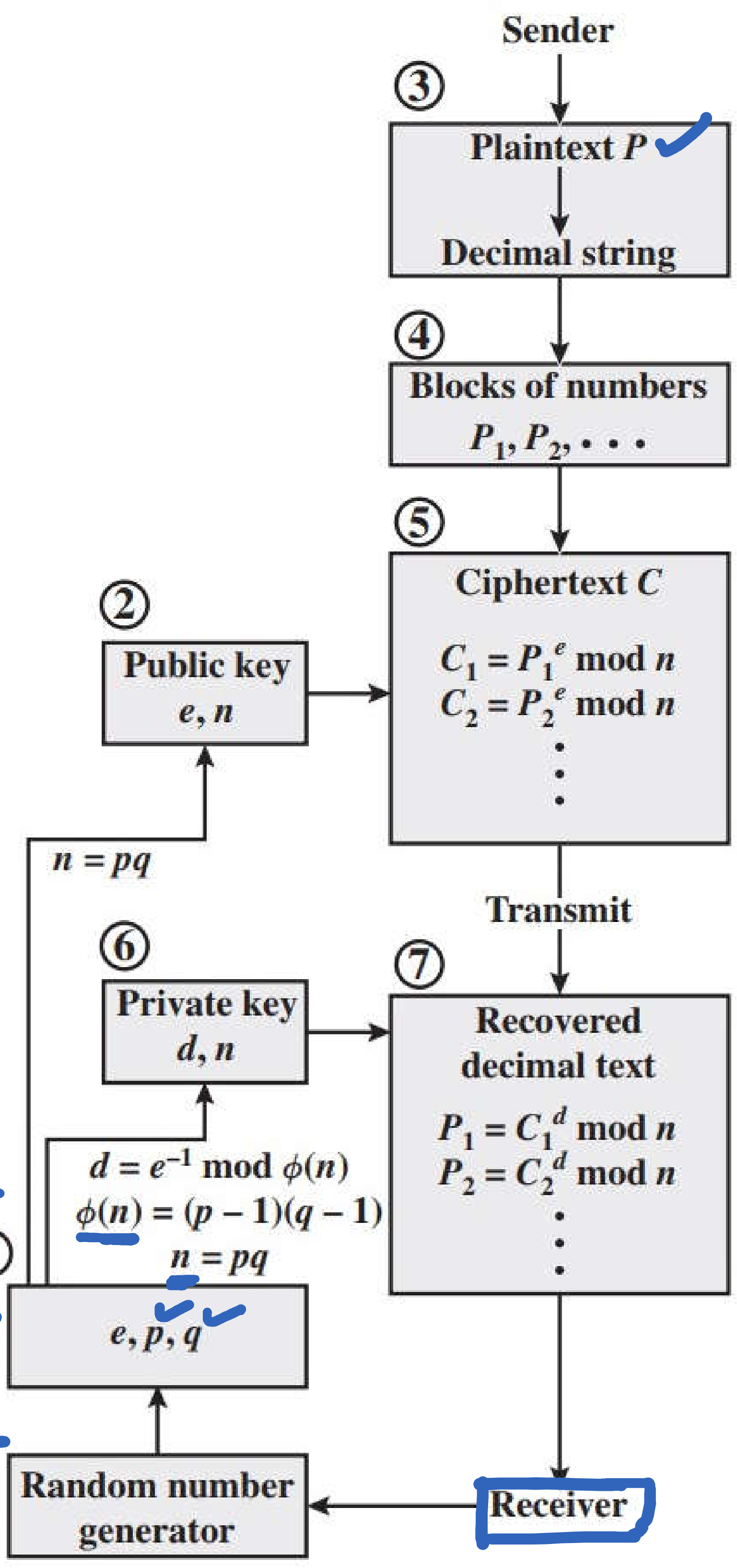
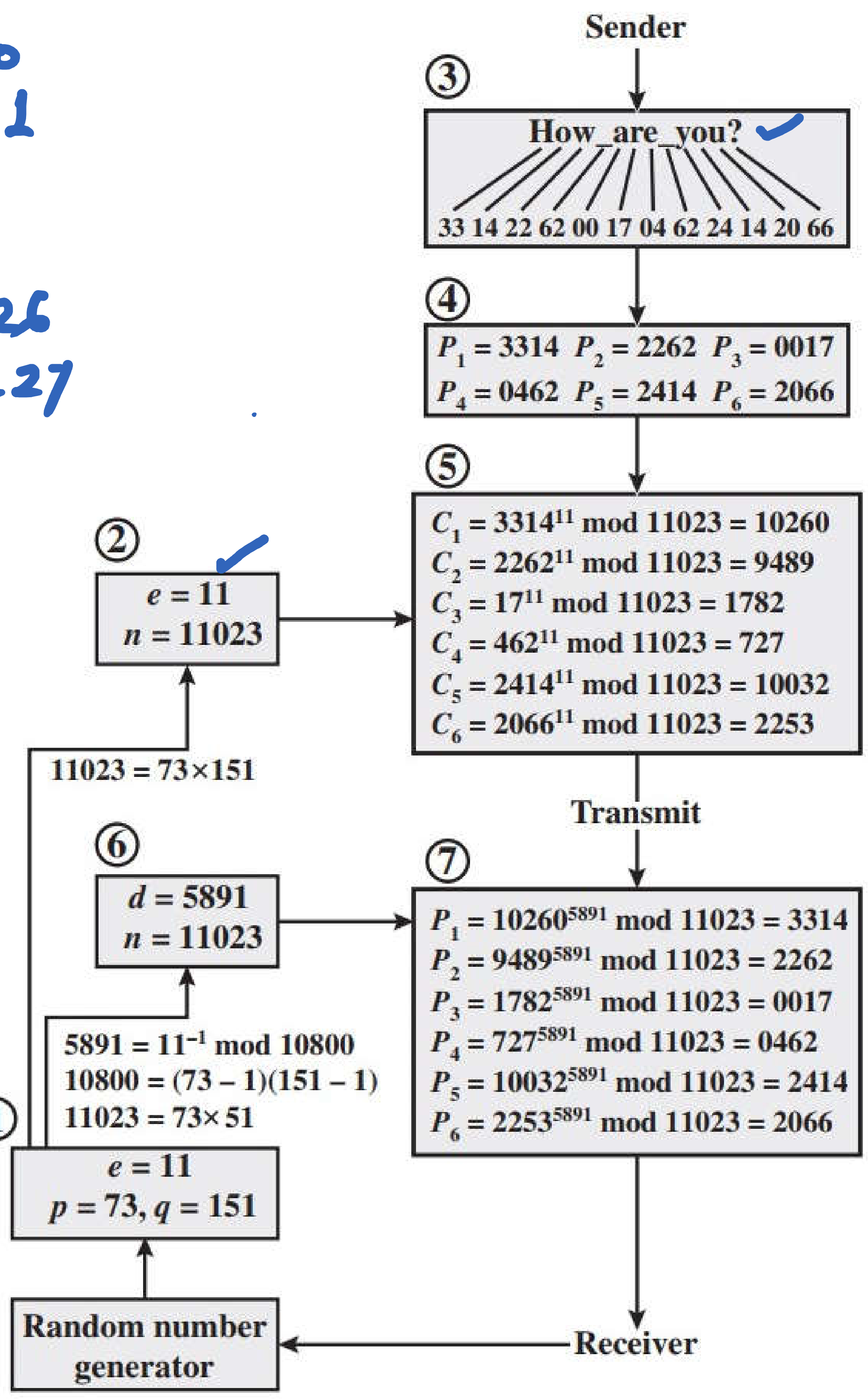
# RSA Cryptosystem




$a = 0$
$b = 1$
⋮
$A = 26$
$B = 27$
⋮

$e \in \{0, 1, 2, \ldots, \phi(n)-1\}$
$\gcd(e, \phi(n)) = 1$




**Remark**: In key generation p & q are very large prime no.

For practical purposes p & q must be at least 512-bits long (p & q should be around 154 decimal digits)

$n \geq 2^{1024}$ (309 decimal digits)

Encryption: $C = P^e \bmod n$
Decryption: $P = C^d \bmod n$

e & d can be very large therefore encryption & decryption would take a long.

Q. How to do the exponentiation quickly?

Ex: $\underbrace{x^8} = \underbrace{x \times x \times x \times \cdots \times x}_{\text{8-times}}$ — 7-operations

✓
$$\left.\begin{aligned} x \cdot x &= x^2 \\ x^2 \cdot x^2 &= x^4 \\ x^4 \cdot x^4 &= x^8 \end{aligned}\right\} \text{3- operations.}$$

$$x^{2^{1024}} = x \times x \times \cdots \times x \quad - \quad 2^{1024} - 1 \text{ operations}$$

$$\left.\begin{aligned} x \cdot x &= x^2 = x^{2^1} \\ x^{2^1} \cdot x^{2^1} &= x^4 = x^{2^2} \\ x^{2^2} \cdot x^{2^2} &= x^8 = x^{2^3} \\ &\vdots \\ x^{2^{1023}} \cdot x^{2^{1023}} &= x^{2^{1024}} \end{aligned}\right\} \text{1024 operations.}$$

What if exponent is not a power of 2?

Square and Multiply Algo. (Binary method or left to right expon.)

$$x^{23} = x^{10111_2}$$

$$\begin{aligned} x^{10_2} &= x \cdot x = x^2 && \text{— Sq.} \\ \left.\begin{aligned} x^{100_2} &= x^2 \cdot x^2 = x^4 \\ x^{101_2} &= x \cdot x^4 = x^5 \end{aligned}\right\} && && \text{— Sq. \& Mult.} \\ \left.\begin{aligned} x^{1010_2} &- x^5 \cdot x^5 = x^{10} \\ x^{1011_2} &- x \cdot x^{10} = x^{11} \end{aligned}\right\} && && \text{— Sq. \& Mult.} \\ \left.\begin{aligned} x^{10110_2} &- x^{11} \cdot x^{11} = x^{22} \\ x^{10111_2} &- x \cdot x^{22} = x^{23} \end{aligned}\right\} && && \text{— Sq. \& Mult.} \end{aligned}$$

Procedure:

1. Convert the exponent to binary.
2. For the first 1, list the base $x$.
3. If the next bit is zero, square the no. obtained in ②
   Else square the no. obtained in ② & mult. by the no. obtained in ②.

Remarks: 1. To speed up the operation of RSA algo. using public key, a specific choice of $e$ is made.

2. The common choice of $e$ is $65537 = 2^{16}+1$.
3. other popular choices are 3 & 17.
4. RSA is vulnerable if $e=3$. (Refer. W. Stalling)

## Security of RSA

Possible Attacks on RSA:

1. Factorization:
2. Chosen Ciphertext
3. Encryption Exp. ($e$)
4. Decryption Exp. ($d$)
5. Plaintext
6. Modulus
7. Implementation (Timing and Power)

1. Factorization: (i) If we are able to factorize $n$

$$n = p \times q$$

$$\phi(n) = (p-1)(q-1)$$

$$d = e^{-1} \bmod \phi(n) \quad \{\because e \text{ is public}$$

(ii) If $\phi(n)$ is determined directly. Then

$$d = e^{-1} \bmod \phi(n) \quad \{\because e \text{ is public}$$

(Equivalent to (i))

(iii) If $d$ is determined directly. { As time consuming as factorizing $n$.

To prevent factoring attack $n$ must be at least 1024-bit number (i.e. 309 decimal digits).

## 2. Chosen ciphertext attack

Assume that:

1. Eve can intercept the message sent by Alice to Bob.
2. Bob will decrypt an arbitrary ciphertext for Eve. (Eve - adversary)

Alice

$C = P^e \bmod n$

Eve

$C$

$Y = C \times X^e \bmod n$

$\underbrace{X \in \mathbb{Z}_n^*}$ is a random number

Bob

$Z = Y^d \bmod n$

$\underline{(\mathbb{Z}_n^*, X)}$

Now,

$$Z = Y^d \bmod n = (C \times X^e)^d \bmod n$$

$$= C^d \times X^{ed} \bmod n$$

$$Z = C^d X \bmod n$$

$$Z = P X \bmod n$$

$$\underbrace{P} = Z \times \underbrace{X^{-1}} \bmod n$$