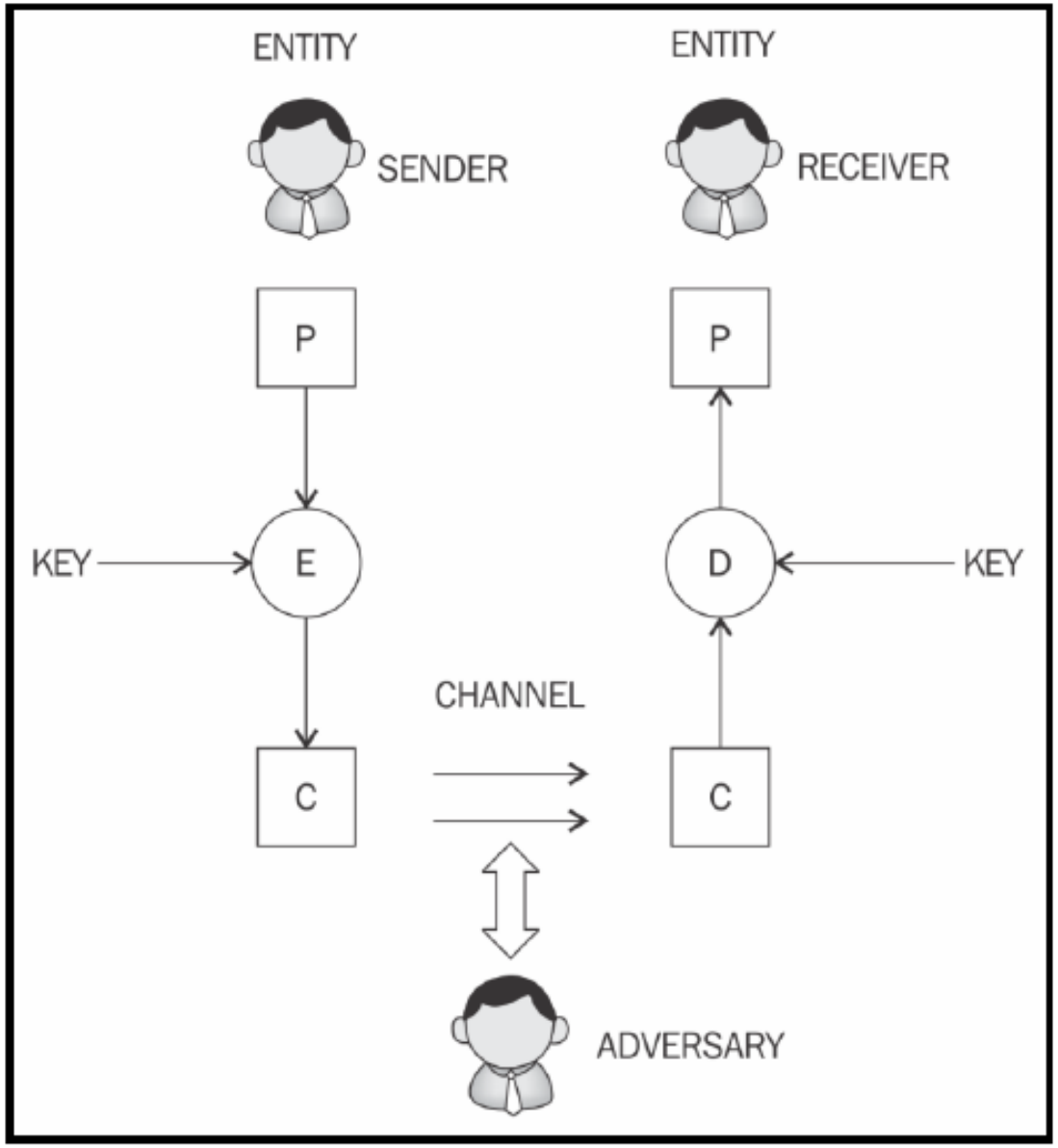
Cryptography and Technical Foundations

Unit-3 (Part-A)

Symmetric Cryptography

Introduction

- **Cryptography** is the science of making information secure in the presence of adversaries.
- **P**, **E**, **C**, and **D** represent plaintext, encryption, ciphertext, and decryption.
- **Entity**: Either a person or system that sends, receives, or performs operations on data
- **Sender**: This is an entity that transmits the data
- **Receiver**: This is an entity that takes delivery of the data
- **Adversary**: This is an entity that tries to circumvent the security service
- **Key**: A key is data that is used to encrypt or decrypt other data
- **Channel**: Channel provides a medium of communication between entities



A model of the generic encryption and decryption model

Mathematics behind Cryptography

- A **set** is a collection of distinct objects, for example, *X = {1, 2, 3, 4, 5}*.
- A **group** is a commutative set with one operation that combines two elements of the set. The group operation is closed and associated with a defined identity element.
- **Closure** (closed) means that if, for example, elements *A* and *B* are in the set, then the resultant element after performing an operation on the elements is also in the set.
- **Associative** means that the grouping of elements does not affect the result of the operation.
- A **field** is a set that contains both additive and multiplicative groups. the **distributive law** is also applied. The law dictates that the same sum or product will be produced even if any of the terms or factors are reordered.
- A **finite field** is one with a finite set of elements. Also known as *Galois fields*, these structures are of particular importance in cryptography as they can be used to produce accurate and error-free results of arithmetic operations.

- The **order** is the number of elements in a field. It is also known as the *cardinality* of the field.
- An **abelian group** is formed when the operation on the elements of a set is commutative. The commutative law means that changing the order of the elements does not affect the result of the operation, for example, *A X B = B X A*.
- A **prime field** is a finite one with a prime number of elements. It has specific rules for addition and multiplication, and each nonzero element in the field has an inverse. Addition and multiplication operations are performed modulo *p*, that is, prime.
- If more than one operation can be defined over an abelian group, that group becomes a **ring**. A ring must have closure and associative and distributive properties.
- A **cyclic group** is a type of group that can be generated by a single element called the *group generator*.
- Also known as clock arithmetic, numbers in modular arithmetic wrap around when they reach a certain fixed number. This fixed number is a positive number called **modulus**, and all operations are performed concerning this fixed number.

Cryptography services

- **Confidentiality** is the assurance that information is only available to authorized entities.
- **Authentication** provides assurance about the identity of an entity or the validity of a message.
  - Entity authentication
  - Data origin authentication
- **Non-repudiation** is the assurance that an entity cannot deny a previous commitment or action by providing incontrovertible evidence. It is a security service that offers definitive proof that a particular activity has occurred.
- **Accountability** is the assurance which states that actions affecting security can be traced back to the responsible party. This is usually provided by logging and audit mechanisms in systems where a detailed audit is required due to the nature of the business.

Entity authentication

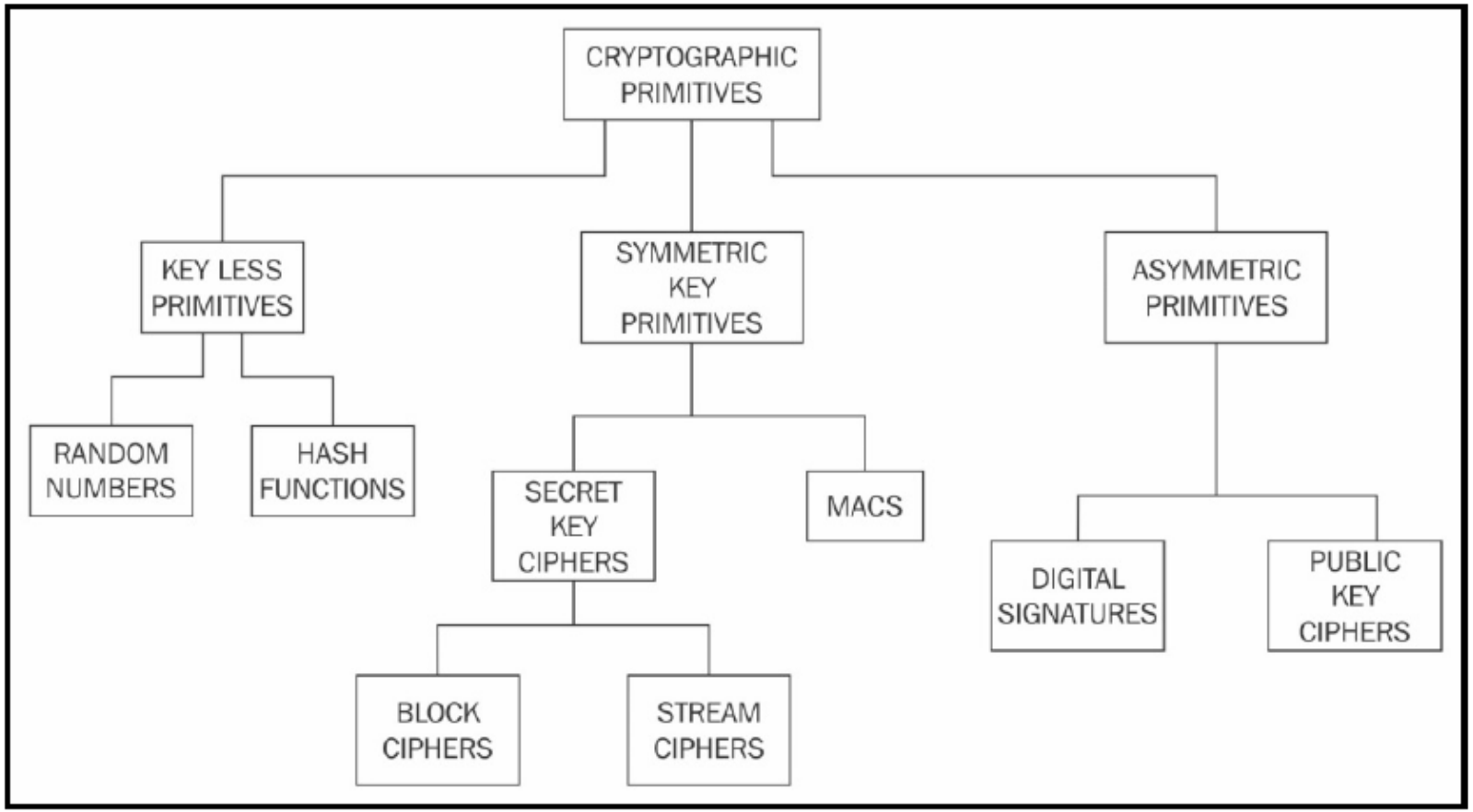
- Traditionally, users are issued a username and password that is used to gain access to the various platforms with which they are working. This practice is known as **single-factor authentication**, as there is only one factor involved, namely, *something you know*, that is, the password and username.
- This type of authentication is not very secure for a variety of reasons, for example, password leakage; therefore, additional factors are now commonly used to provide better security. The use of additional techniques for user identification is known as **multifactor authentication** (or two-factor authentication if only two methods are used).
  - The first factor is *something you have*, such as a hardware token or a smart card. In this case, a user can use a hardware token in addition to login credentials to gain access to a system. the login password is also used in conjunction with the hardware token.
  - The second factor is *something you are*, which uses biometric features to identify the user.

Data origin authentication

- Also known as *message authentication*, **data origin authentication** is an assurance that the source of the information is indeed verified. Data origin authentication guarantees data integrity because if a source is corroborated, then the data must not have been altered.
  - Message Authentication Codes (MACs)
  - Digital signatures

Cryptographic primitives

- **Cryptographic primitives** are the basic building blocks of a security protocol or system.
- A **security protocol** is a set of steps taken to achieve the required security goals by utilizing appropriate security mechanisms. Various types of security protocols are in use, such as authentication protocols, non-repudiation protocols, and key management protocols.

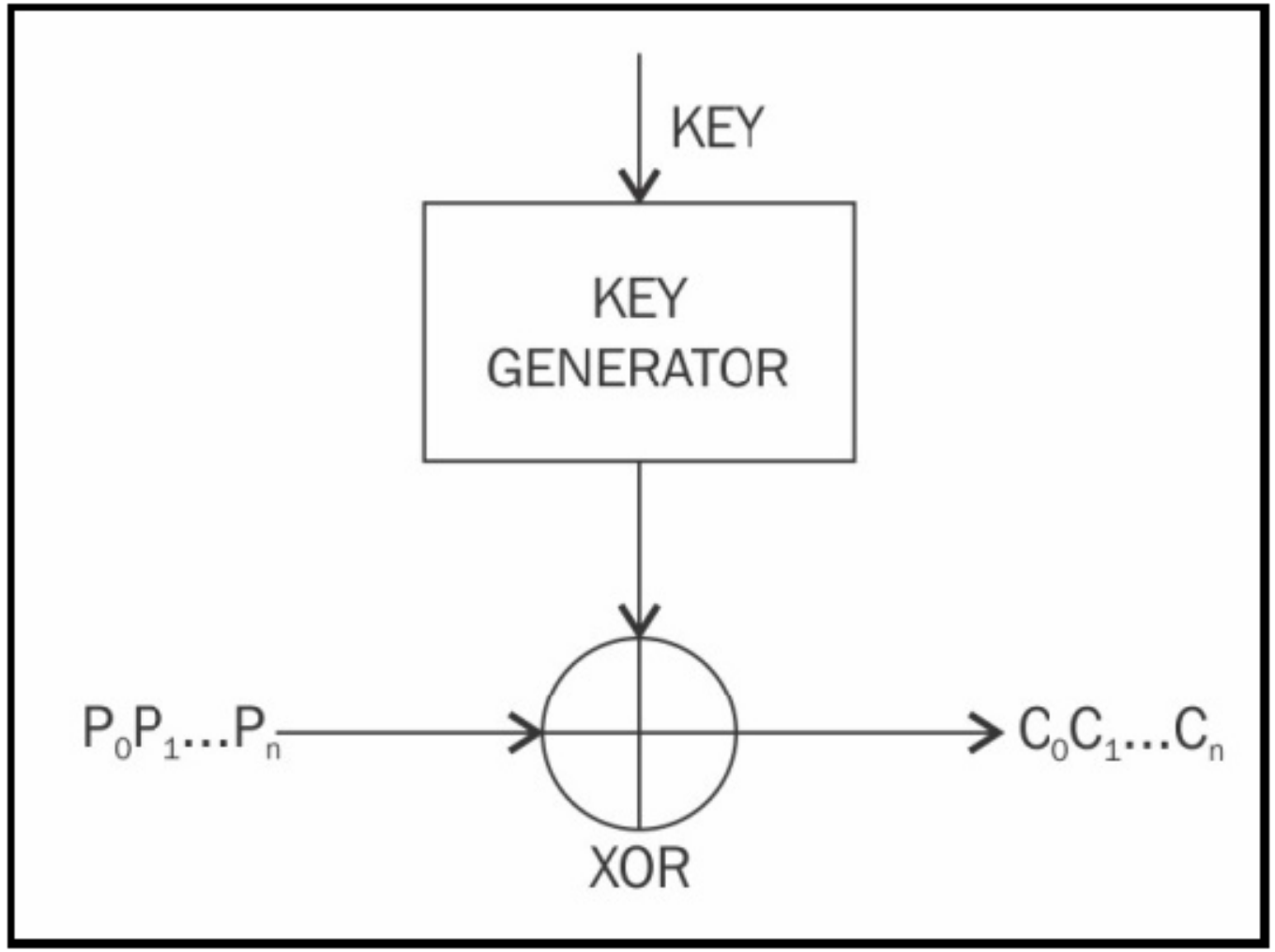


Symmetric cryptography

- **Symmetric cryptography** refers to a type of cryptography where the key that is used to encrypt the data is the same one that is used for decrypting the data. Thus, it is also known as **shared key cryptography**.
- The key must be established or agreed upon before the data exchange occurs between the communicating parties. This is the reason it is also called **secret key cryptography**.
- There are two types of symmetric ciphers:
  - ***Stream ciphers*** *ex:* RC4 and A5
  - ***Block ciphers*** *ex:* Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

Stream ciphers

- **Stream ciphers** are encryption algorithms that apply encryption algorithms on a bit-by-bit basis (one bit at a time) to plaintext using a keystream.
- There are two types of stream ciphers:
  - **Synchronous stream ciphers** are those where the keystream is dependent only on the key
  - **Asynchronous stream ciphers** have a keystream that is also dependent on the encrypted data
- In stream ciphers, encryption and decryption are the same function because they are simple modulo-2 additions or XOR operations.
- The fundamental requirement in stream ciphers is the security and randomness of keystreams. Various techniques ranging from pseudorandom number generators to true random number generators implemented in hardware have been developed to generate random numbers

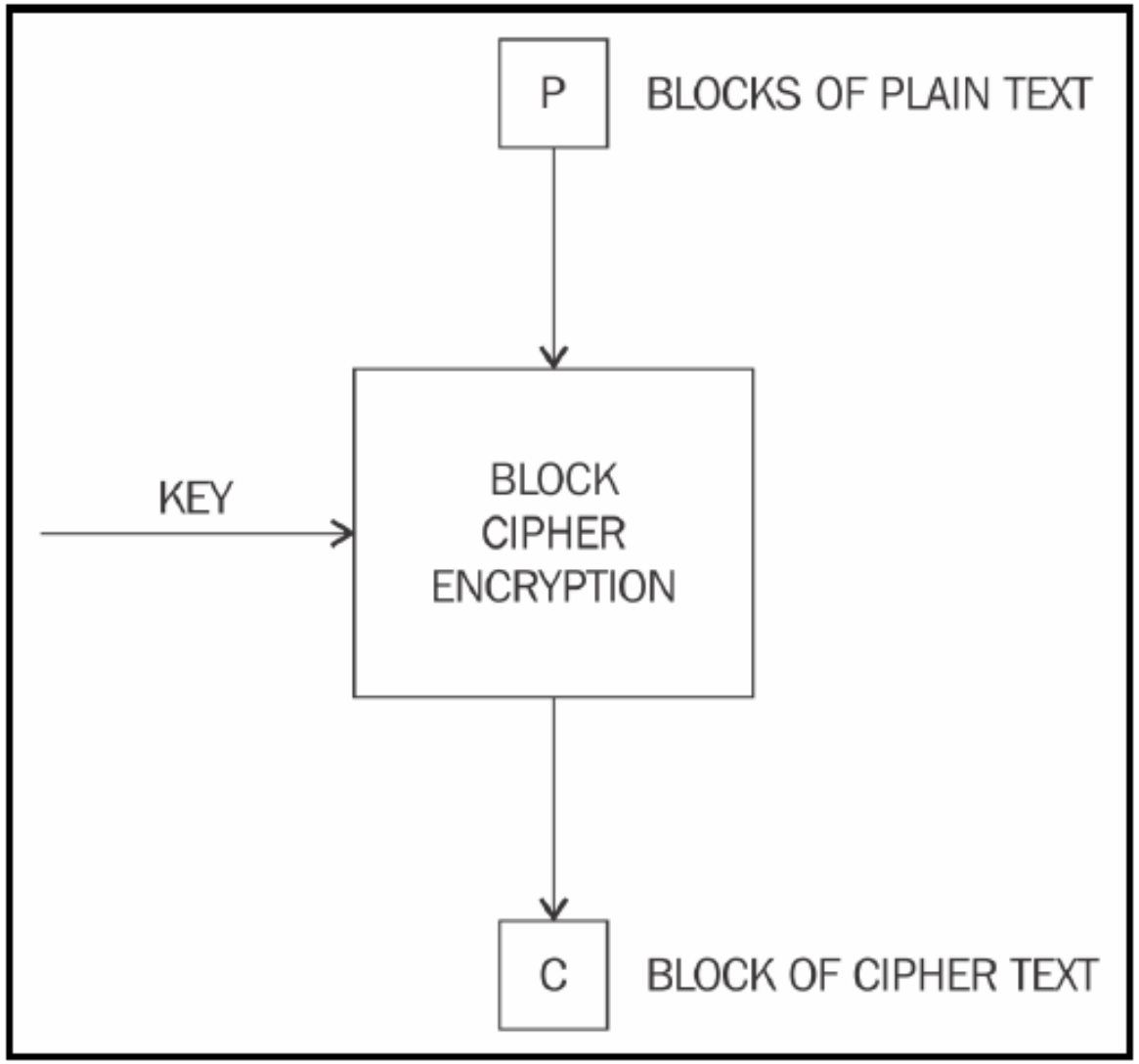


Operation of a stream cipher

Block ciphers

- **Block ciphers** are encryption algorithms that break up the text to be encrypted (plaintext) into blocks of a fixed length and apply the encryption block-by-block. Block ciphers are generally built using a design strategy known as a **Feistel cipher**.
- Recent block ciphers, such as AES (Rijndael) have been built using a combination of substitution and permutation called a **Substitution-Permutation Network** (**SPN**).
- This structure is based on the idea of combining multiple rounds of repeated operations to achieve desirable cryptographic properties known as *confusion* and *diffusion*.
  - Confusion makes the relationship between the encrypted text and plaintext complex. This is achieved by substitution. In modern cryptographic algorithms, substitution is performed using lookup tables called *Sboxes*. In practice, this is achieved by transposition or permutation.
  - The diffusion property spreads the plaintext statistically over the encrypted data. This ensures that even if a single bit is changed in the input text, it results in changing at least half (on average) of the bits in the ciphertext.
- Feistel networks operate by dividing data into two blocks (left and right) and processing these blocks via keyed *round functions* in iterations to provide sufficient pseudorandom permutation.

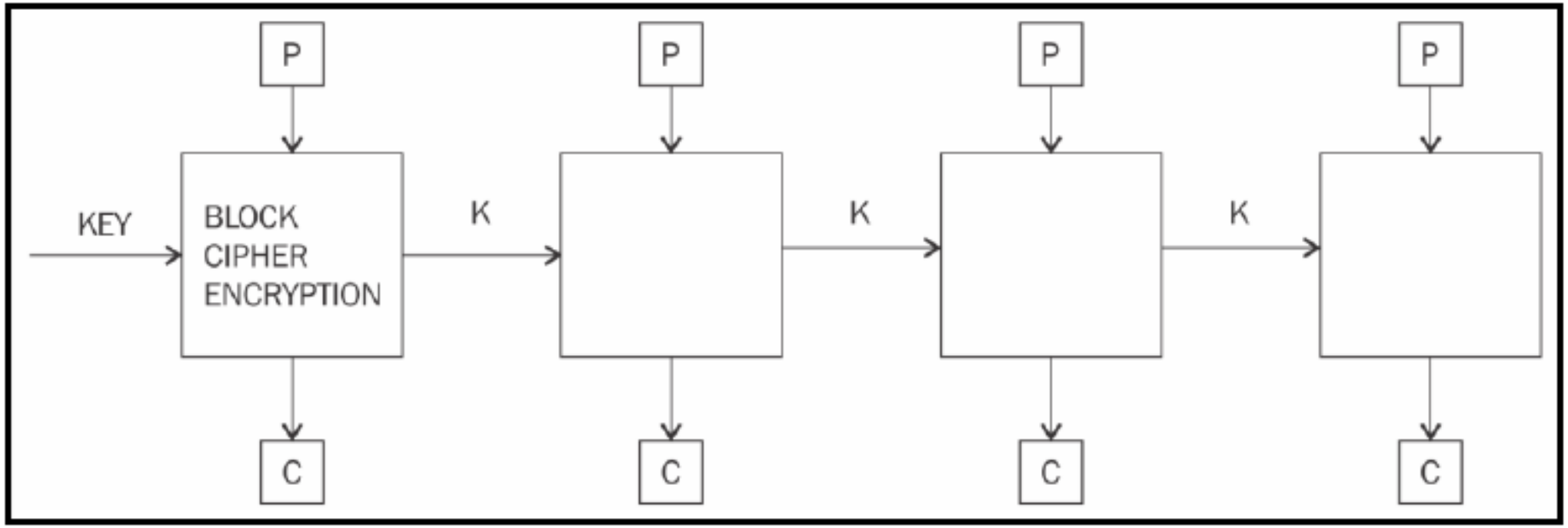
- **Block encryption mode**
- In **block encryption mode**, the plaintext is divided into blocks of fixed length depending on the type of cipher used. Then the encryption function is applied to each block.
- Various modes of operation for block ciphers are
  - **Electronic Code Book** (**ECB**),
  - **Cipher Block Chaining** (**CBC**),
  - **Output Feedback** (**OFB**) mode, and
  - **Counter** (**CTR**) mode.
- These modes are used to specify the way in which an encryption function is applied to the plaintext.



Simplified operation of a block cipher

Electronic Code Book

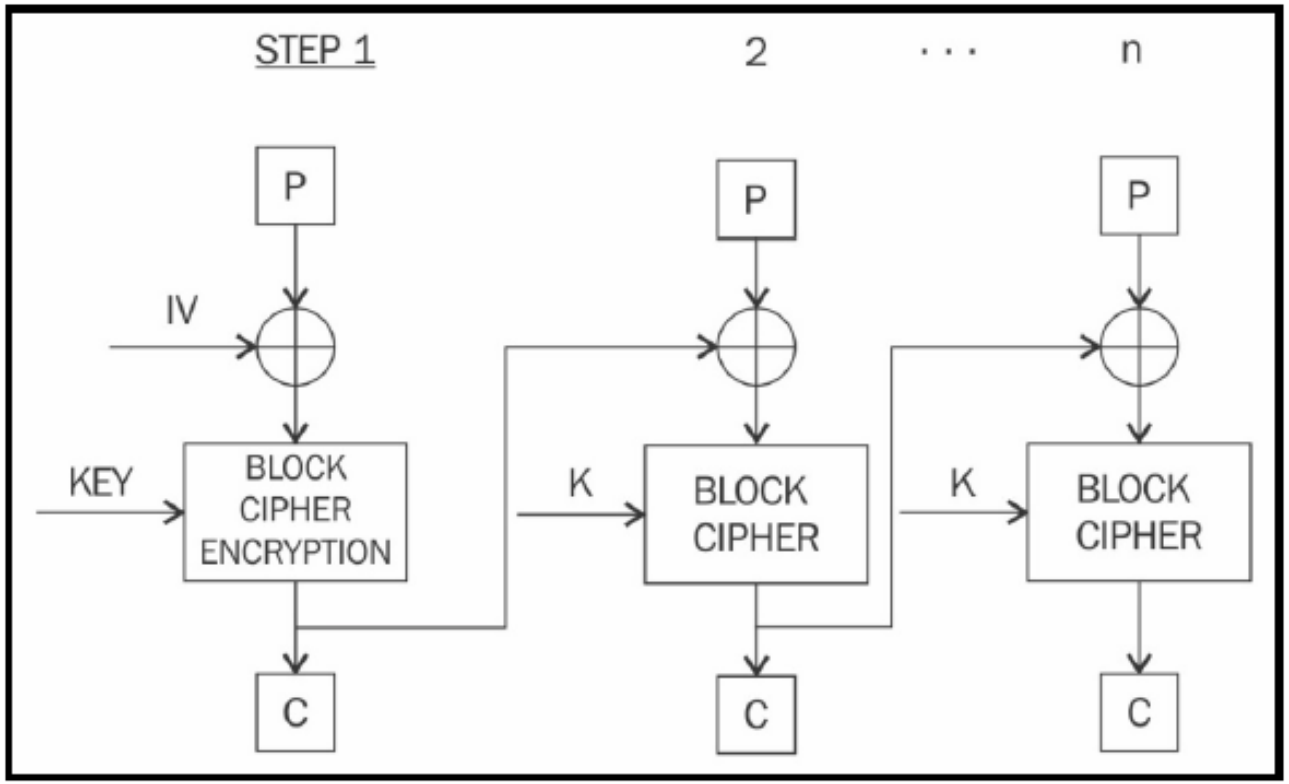
- **Electronic Code Book** (**ECB**) is a basic mode of operation in which the encrypted data is produced as a result of applying the encryption algorithm one-by-one to each block of plaintext.
- This is the most straightforward mode, but it should not be used in practice as it is insecure and can reveal information.



Electronic Code Book mode for block ciphers

Cipher Block Chaining

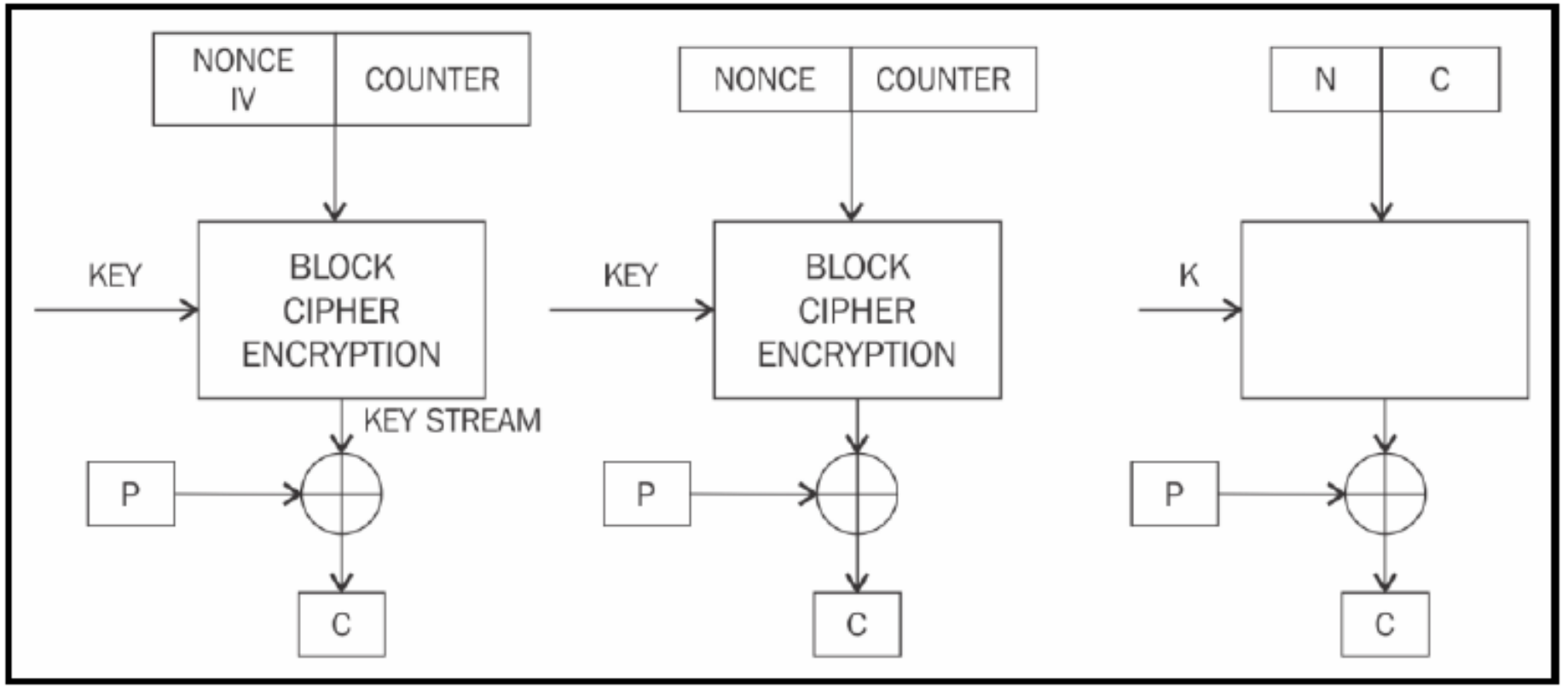
- In **Cipher Block Chaining** (**CBC**) mode, each block of plaintext is XOR'd with the previously-encrypted block.
- CBC mode uses the **Initialization Vector** (**IV**) to encrypt the first block. It is recommended that the IV be randomly chosen



Cipher block chaining mode

Counter mode

- The **Counter** (**CTR**) mode effectively uses a block cipher as a stream cipher.
- In this case, a unique nonce is supplied that is concatenated with the counter value to produce a **keystream**



Counter mode

Keystream generation mode

- In **keystream generation mode**, the encryption function generates a keystream that is then XOR'd with the plaintext stream to achieve encryption.

Message authentication mode

- In **message authentication mode**, a **Message Authentication Code** (**MAC**) results from an encryption function.
- The MAC is a cryptographic checksum that provides an integrity service. The most common method to generate a MAC using block ciphers is CBC-MAC, where a part of the last block of the chain is used as a MAC.
- The resultant message and MAC of the message once received by the receiver can be checked by encrypting the message received again by the key and comparing it with the MAC received from the sender. If they both match, then the message has not modified by unauthorized user thus integrity service is provided.

Cryptographic hash mode

- Hash functions are primarily used to compress a message to a fixed-length digest. In **cryptographic hash mode**, block ciphers are used as a compression function to produce a hash of plaintext.

Data Encryption Standard

- The **Data Encryption Standard** (**DES**) was introduced by the U.S. **National Institute of Standards and Technology** (**NIST**) as a standard algorithm for encryption.
- DES uses a key of only 56 bits, which raised some concerns.
- This problem was addressed with the introduction of **Triple DES** (**3DES**), which proposed the use of a 168-bit key by means of three 56-bit keys and the same number of executions of the DES algorithm, thus making brute force attacks almost impossible.
- However, other limitations, such as slow performance and 64-bit block size, were not desirable.

Advanced Encryption Standard

- Rijndael invented by cryptographers Joan Daemen and Vincent Rijmen was standardized as **Advanced Encryption Standard** (**AES**) with minor modifications by NIST.
- So far, no attack has been found against AES that is more effective than the brute-force method.
- In the AES standard, however, only a 128-bit block size is allowed. However, key sizes of 128-bit, 192-bit, and 256-bit are permissible.

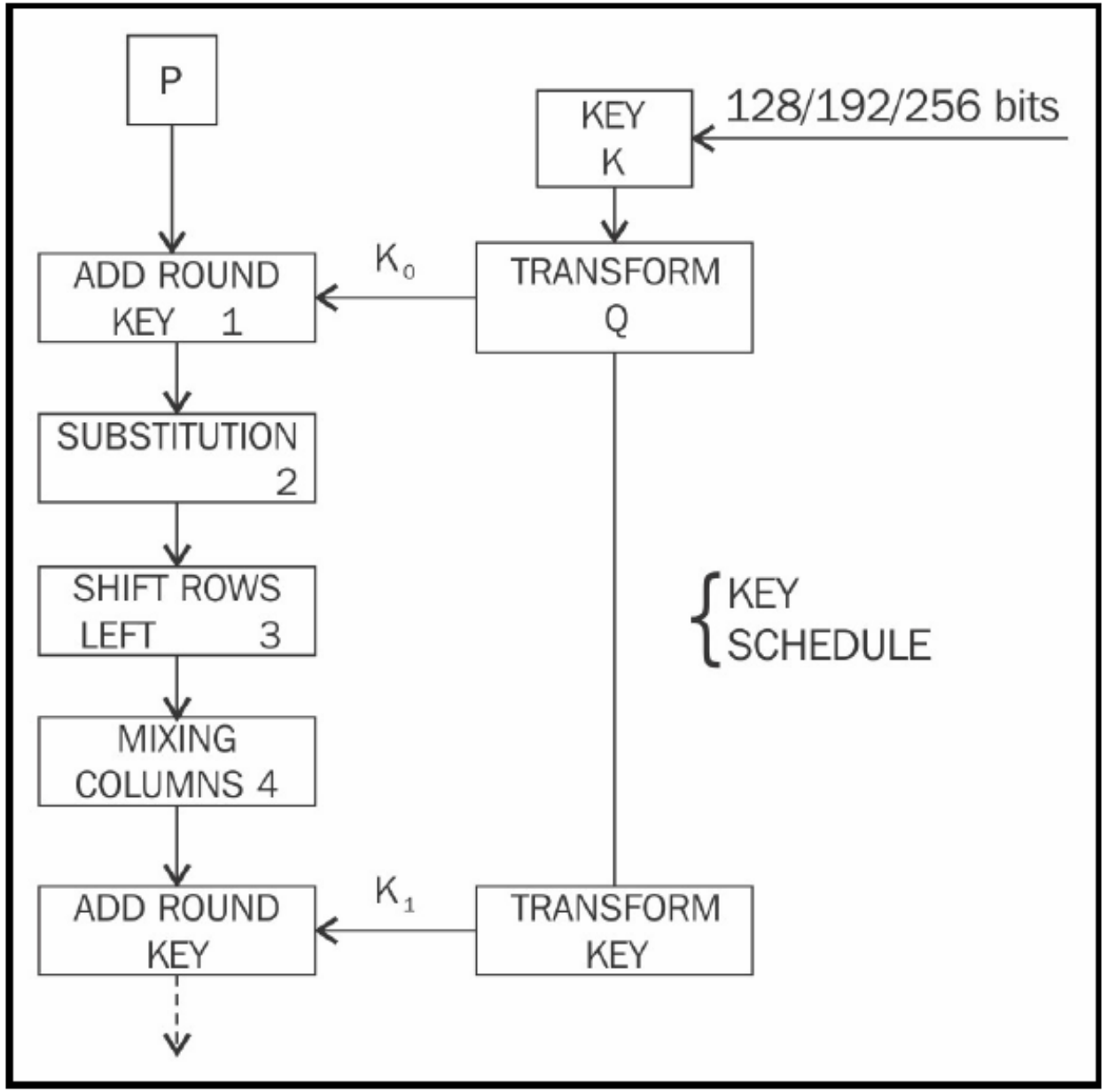
How AES works

- During AES algorithm processing, a 4 x 4 array of bytes known as the **state** is modified using multiple rounds.
- Full encryption requires 10 to 14 rounds, depending on the size of the key.
- Once the state is initialized with the input to the cipher, four operations are performed in four stages to encrypt the input.

Key size	Number of rounds required
128-bit	10 rounds
192-bit	12 rounds
256-bit	14 rounds

- These stages are:

1. In the AddRoundKey step, the state array is XOR'd with a subkey, which is derived from the master key
2. SubBytes is the substitution step where a lookup table (S-box) is used to replace all bytes of the state array
3. The ShiftRows step is used to shift each row to the left, except for the first one, in the state array to the left in a cyclic and incremental manner
4. Finally, all bytes are mixed in the MixColumns step in a linear fashion, columnwise

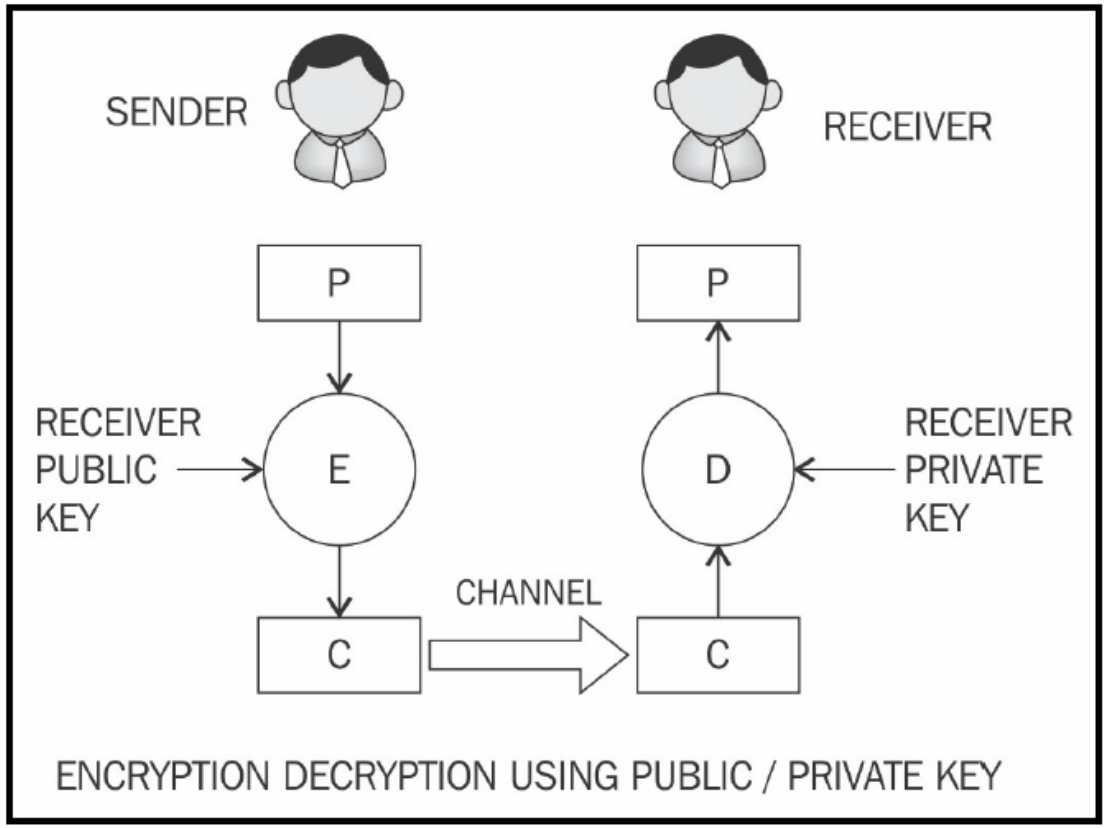


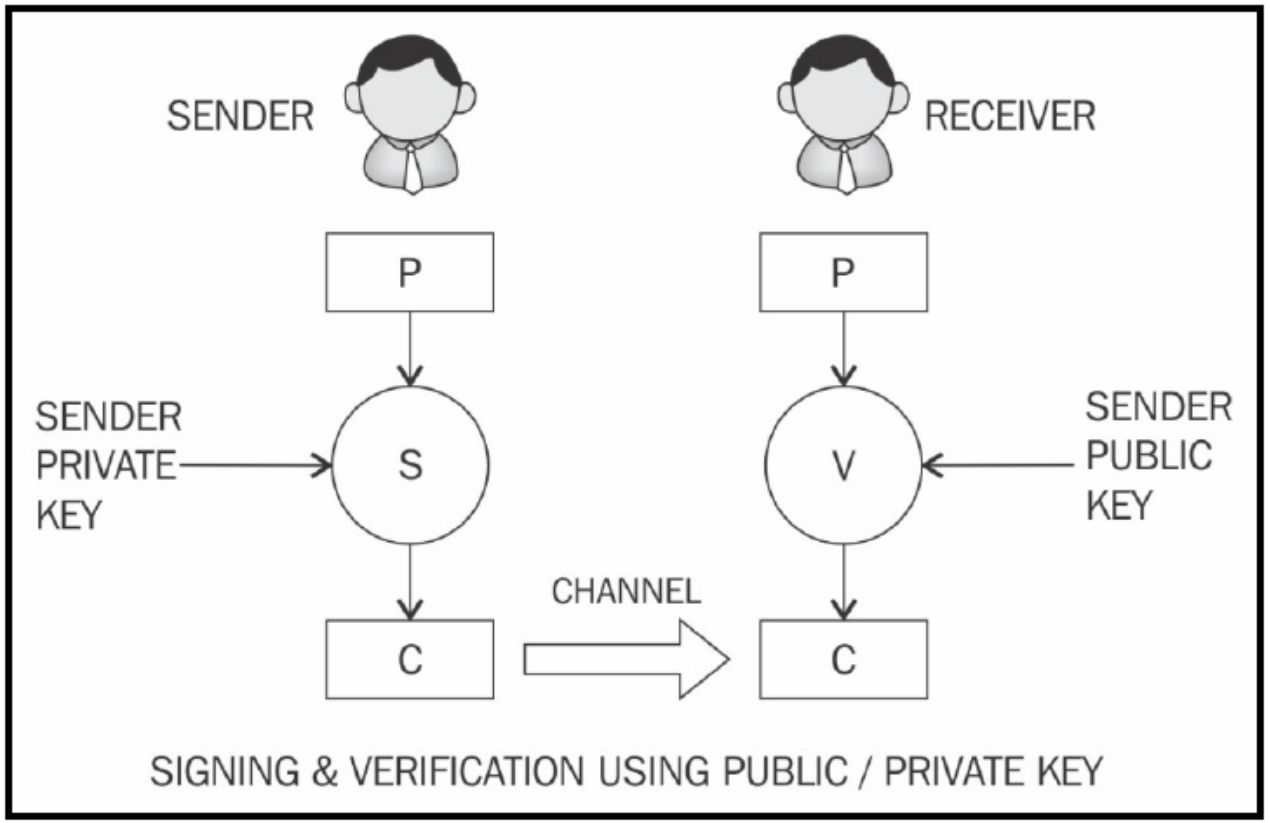
AES block diagram, showing the first round of AES encryption. In the last round, the mixing step is not performed

Asymmetric Cryptography

Asymmetric cryptography

- **Asymmetric cryptography** refers to a type of cryptography where the key that is used to encrypt the data is different from the key that is used to decrypt the data.
- This is also known as **public key cryptography**. It uses both public and private keys to encrypt and decrypt data, respectively.





- Security mechanisms offered by public key cryptosystems include key establishment, digital signatures, identification, encryption, and decryption.
- **Key establishment mechanisms** are concerned with the design of protocols that allow the setting up of keys over an insecure channel.
- Non-repudiation services, a very desirable property in many scenarios, can be provided using **digital signatures**.
- It is important not only to authenticate a user but also to identify the entity involved in a transaction. This can also be achieved by a combination of digital signatures and **challenge response protocols**.
- The encryption mechanism to provide confidentiality can also be obtained using public key cryptosystems, such as RSA, ECC, and ElGammal.
- Public key algorithms are slower in terms of computation than symmetric key algorithms. Therefore, they are not commonly used in the encryption of large files or the actual data that requires encryption.

Mathematical functions of Asymmetric mechanisms

- Public key cryptography algorithms are based on various underlying mathematical functions. The three main categories of asymmetric algorithms are described here.

Integer factorization

- **Integer factorization schemes** are based on the fact that large integers are very hard to factor. RSA is the prime example of this type of algorithm.

Discrete logarithm

- A **discrete logarithm scheme** is based on a problem in modular arithmetic. It is easy to calculate the result of modulo function, but it is computationally impractical to find the exponent of the generator.

For example: $3^2 \bmod 10 = 9$

- Now, given *9*, the result of the preceding equation finding *2* which is the exponent of the generator *3* in the preceding question, is extremely hard to determine.
- Diffie-Hellman key exchange and digital signature algorithms uses this technique.

Elliptic curves

- The **elliptic curves algorithm** is based on the discrete logarithm problem discussed earlier but in the context of elliptic curves. An **elliptic curve** is an algebraic cubic curve over a field, which can be defined by the following equation. The curve is non-singular, which means that it has no cusps or self-intersections. It has two variables *a* and *b*, as well as a point of infinity.

$$y^2 = x^3 + ax + b$$

- Here, *a* and *b* are integers whose values are elements of the field on which the elliptic curve is defined. Elliptic curves can be defined over real numbers, rational numbers, complex numbers, or finite fields. For cryptographic purposes, an elliptic curve over prime finite fields is used instead of real numbers. Additionally, the prime should be greater than 3. Different curves can be generated by varying the value of *a* and/or *b*.
- **Elliptic Curve Digital Signature Algorithm** (**ECDSA**) and the **Elliptic Curve Diffie-Hellman** (**ECDH**) key exchange are commonly used mechnisms.

Rivest–Shamir–Adleman (RSA)

- This type of public key cryptography is based on the integer factorization problem, where the multiplication of two large prime numbers is easy, but it is difficult to factor it (the result of multiplication, product) back to the two original numbers.
- An RSA key pair is generated by performing the following steps:

1. **Modulus generation**:
   - Select *p* and *q*, which are very large prime numbers
   - Multiply *p* and *q*, *n=p.q* to generate modulus *n*
2. **Generate co-prime**:
   - Assume a number called *e*.
   - *e* should satisfy a certain condition; that is, it should be greater than *1* and less than *(p-1) (q-1)*. In other words, *e* must be a number such that no number other than *1* can divide *e* and *(p-1) (q-1)*. This is called **co-prime**, that is, *e* is the co-prime of *(p-1) (q-1)*.
3. **Generate the public key**:
   - The modulus generated in step 1 and co-prime *e* generated in step 2 is a pair together that is a public key. This part is the public part that can be shared with anyone; however, *p* and *q* need to be kept secret.
4. **Generate the private key**:
   - The private key, called *d* here, is calculated from *p*, *q*, and *e*. The private key is basically the inverse of *e modulo (p-1) (q-1)*. In the equation form, it is this as follows:

$$ed = 1 \bmod (p-1)(q-1)$$