Lesson 003: Cypher Primitives

The Basic Building blocks of encryption

Recap on Encryption Systems

- Keyspace
  - Total number of possible values of keys in a crypto algorithm
- Substitution Cipher
  - Convert one letter to another
- Cryptoquip
- Transposition Cipher
  - Change position of letter in text
- Word Jumble
- Monoalphabetic Cipher
- Caesar

Recap on Encryption Systems

- Polyalphabetic Cipher
  - Vigenère
- Modular Mathematics
  - Running Key Cipher
- One-time Pads
  - Randomly generated keys

Attributes of Strong Encryption

- **Confusion**
  - Change key values each round
  - Performed through substitution
  - Complicates plaintext/key relationship
- **Diffusion**
  - Change location of plaintext in ciphertext
  - Done through transposition

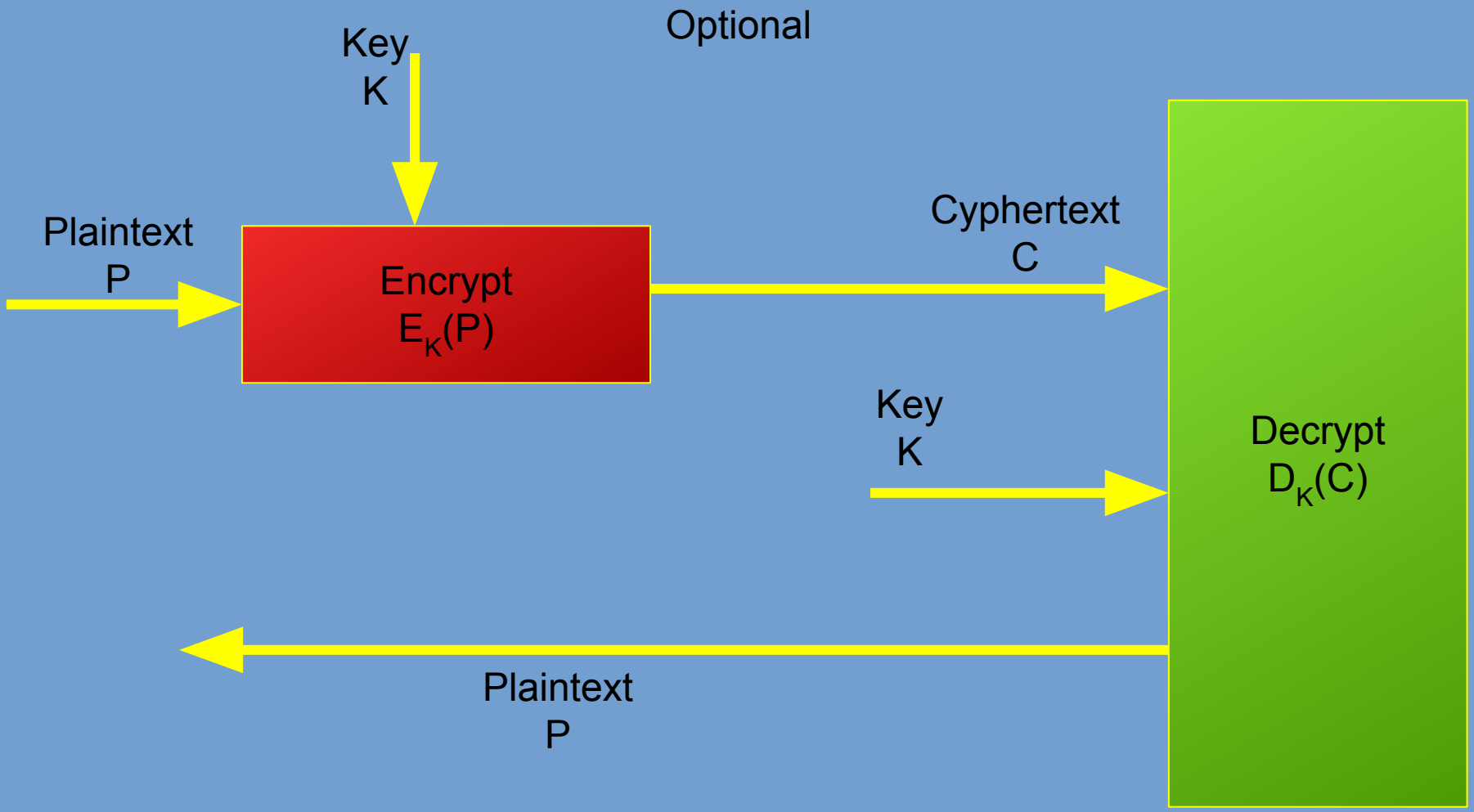
General Mathematical/Formal Representation

- Key = **K**
- Plaintext = **P**
- Cyphertext = **C**
- Encrypt = **E()**
- Decrypt = **D()**

General Mathematical/Formal Representation

- Encryption Process
  - $C = C_K = E_K(P)$
    - C_K just means Cyphertext using Key **K**
    - $E_K(P)$ just means Encrypt Plaintext **P** using Key **K**
- Decryption Process
  - $P = D_K(C) = D_K(E_K(P))$
    - $D_K(C)$:just means Decrypt Cyphertext **C** using Key **K**

Typical Encryption/Decryption Process



Asymmetric cyphers uses different keys for Encrypting and Decrypting
while Symmetric cyphers just use the same key for both processes

- Open up an encryption “box”, break it apart to reveal it's basic operations and what you get is a set of **Boolean logic** and **basic arithmetic operations** combined in an elaborate elegant and creative sequence.

- Lets cover the building blocks.

Boolean Logic

- Boolean algebra is the branch of algebra in which the values of the variables are the truth values true and false, usually denoted 1 and 0 respectively.
- 1 represents ON or TRUE
- 0 represents OFF or FALSE

AND Logic

- The output will only be a 1 when ALL inputs are 1 otherwise it will remain a 0
- Y=A.B

INPUTS		OUTPUT
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Logic

- The output will be a 1 as long as at least one of it's inputs is a 1.
- Y=A+B

INPUTS		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT Logic

- Negates the input
- 0 becomes 1 and vice versa
- If A is the input then the output becomes !A

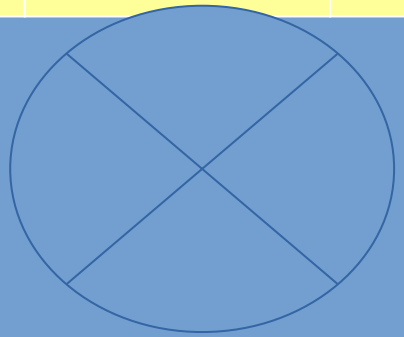
INPUT	OUTPUT
A	A
0	1
1	0

XOR Logic Gate

- The Output will be a 1 as long as the inputs are different
- If input A=B then Y = 0 else Y = 1

INPUTS		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

Symbol



Substitution

- Encoding where units of a plaintext data are replaced with cyphertext according to a FIXED system
- May be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth.
- The receiver deciphers the text by performing the inverse substitution.

Substitution example

Pt	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ct	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

- The message
  - “MEETING AT NOON”
- Translates to
  - “DTTZOFU QZ FGGF”

Transposition

- A valid rule of replacement that permits one to switch a unit of a plaintext data with an adjacent (*one that follows or precedes*) unit of the same plaintext.
- Example
  - “MEETING AT NOON”
  - Transposition >>2
    - >> Meaning value of the plaintex at position *i* will be replaced with the value of the 2nd to the right from *i*.
- Translates to
  - “ETING AT NOONME”

Bit Padding

- This is the process where a chunk that is less than the desired size is added extra bits inorder to reach the required size.
- e.g
  - Bit size requirements is 10 and the chunk size is 3
  - 011
  - Porcedure
    - Add a 1 as the most significant bit to become
      - 1011
    - And add as may 0s as necessary to reach the desired number of bits
      - 0000001011

Example 1

- Develop an encryption algorithm that does the following:
  - Uses a single 256 key
  - Split the data into chunks of 256bits
  - Substitutes the data chunk units using the table below
  - XOR the chunks using the Cypher Key
  - Transposes the data chunks by 2 positions to the right
  - Repeats it 10 times

Substitution Key

	PA	PB	PC	PD
1	0000	0001	0010	0011
2	0100	0101	0110	0111
3	1000	1001	1010	1011
4	1100	1101	1110	1111

Plaintext Table

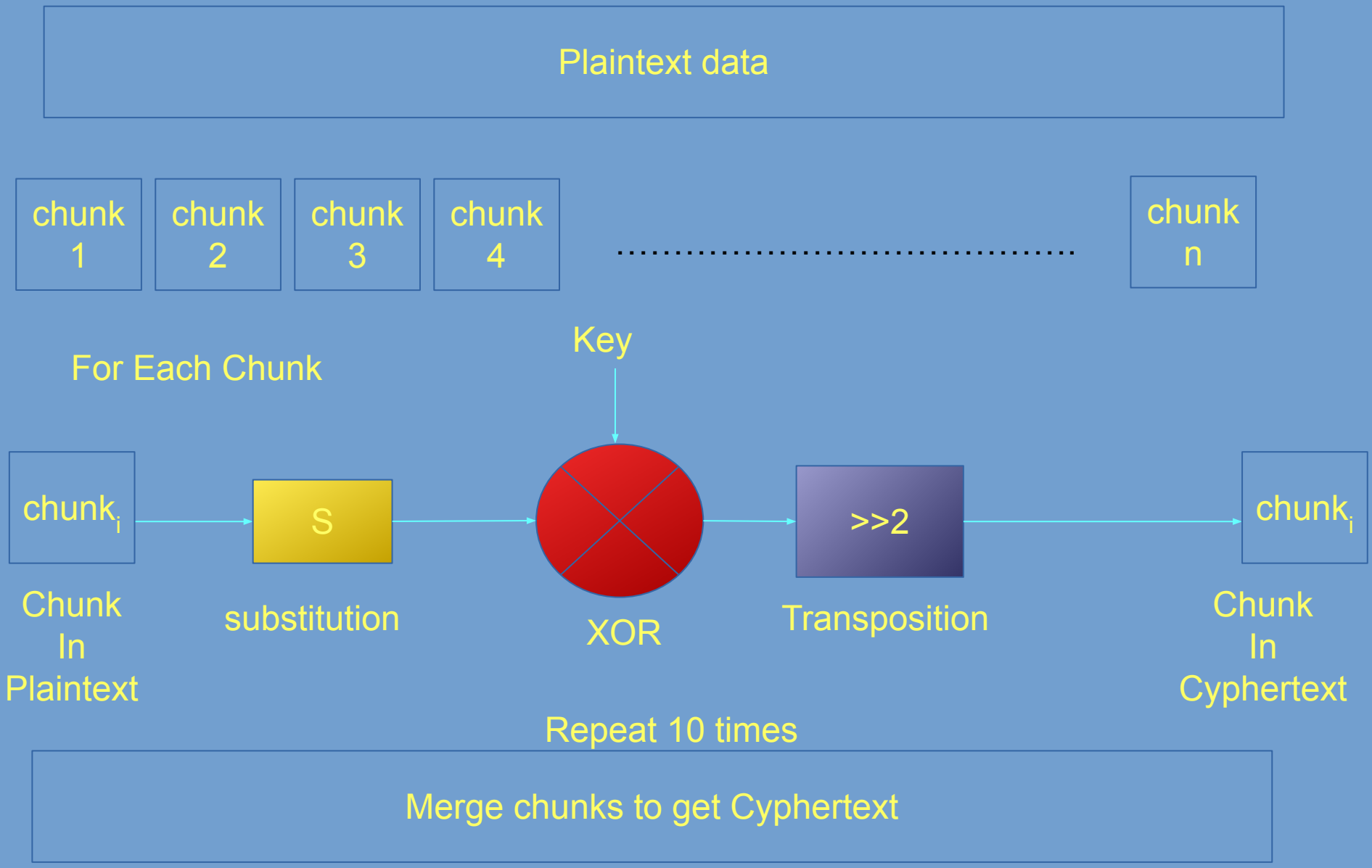
	CA	CB	CC	CD
1	1100	1101	1110	1111
2	1000	1001	1010	1011
3	0100	0101	0110	0111
4	0000	0001	0010	0011

Cypher Table

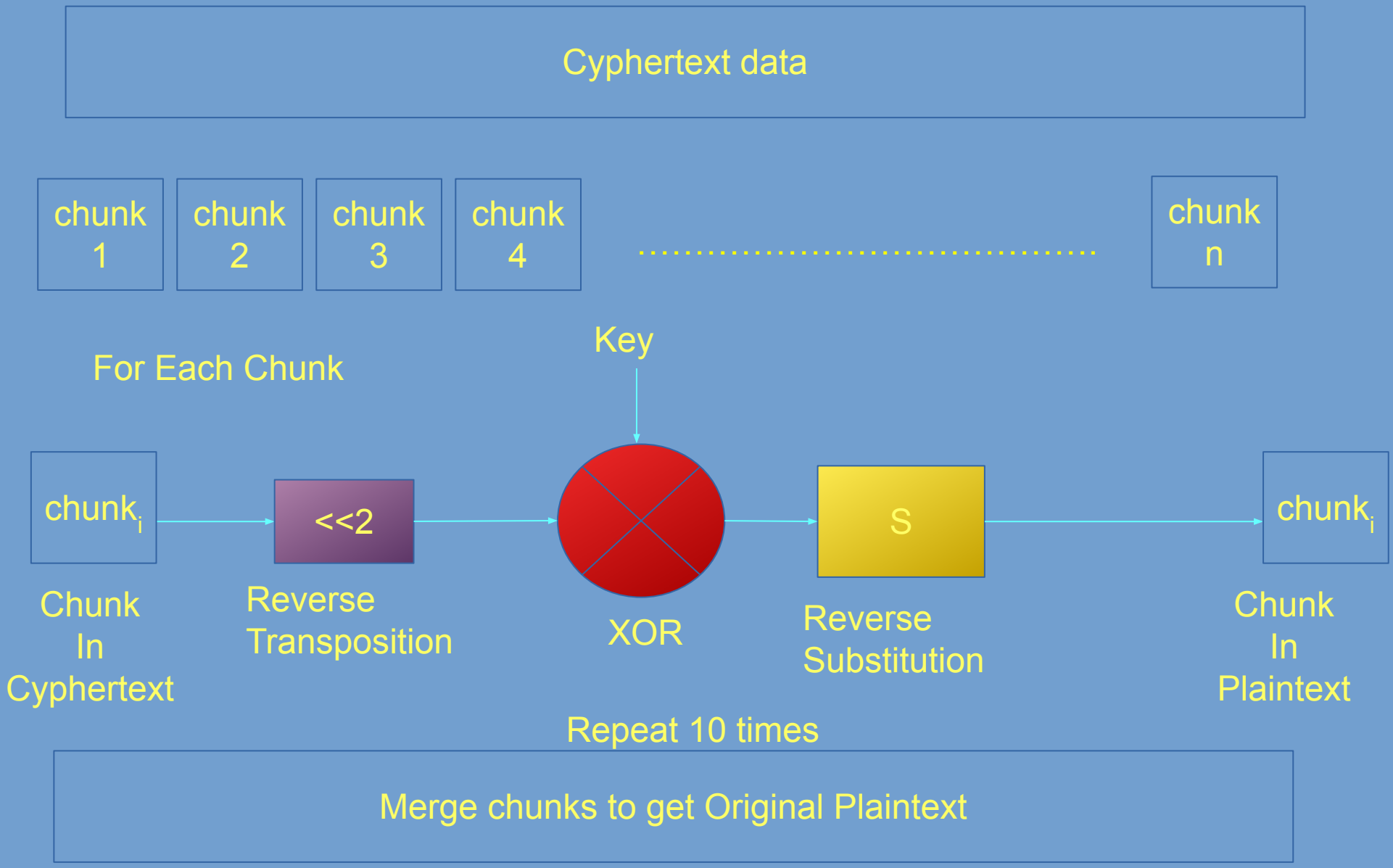
PS: Remember is the data size isn't divisible by 4 then apply padding to the remaining data left after the splits

S will be the symbol of substitution

Solution Diagram Encryption



Solution Diagram Decryption



- **Encryption**
- Assuming the key size and data size is only four bits
- If original data is
- P = 1010 ,Key K = 1101
- Then outcome will be
  - Substitution
    - 1010 => 0110
  - XOR

Chunk bit	Key	Output XOR
0110	1101	1011

  - Transposition >>2
    - 1011 =>1110
- So final bit chunk after encryption will be:
  - 1110

- **Decryption**
- Assuming the key size and data size is only four bits
- If Cypher data is
- P = 1110 ,Key K = 1101
- Then outcome will be
  - Reverse Transposition <<2
    - 1110=>1011
  - XOR

Chunk bit	Key	Output XOR
1011	1101	0110

  - Reverse Substitution
    - 0110 =>1010
- So final bit chunk after decryption will be:
  - 1010

Symbolic Representation

- **Encryption**
- $C=F_K(P)=(T(S(P) \otimes K),2))$
  - Where
  - T(),2 : Transposition of bits By two positions right
  - S(P) : Substitution of Plaintext P
  - : XOR The key and Plaintext
- **Decryption**
- $P=F^{-1}{}_K(C)=(S^{-1}(T^{-1}(C),-2) \otimes K)))$
  - Where
  - T^{-1}(),2 : Inverse Transposition of bits By two positions Left
  - S(C) : Inverse Substitution of Cyphertext C
  - : XOR The key and Cyphertext

Conclusion

- In some algorithms, The larger the Keysize the stronger the encryption

Next. Asymmetric Cryptography