PR Assignment - 01

Aim:- Implementation of Diffe-Hellman key exchange algorithm.

Theory:-

Encryption:- Encryption attempts to make information unreadable by anyone who is not explicitly authorized to view that data. Encryption involves converting human readable plaintext into incomprehensible text, which is known as ciphertext. Essentially, this means taking readable data & changing it so that it appears random.

Types of Encryption:- There are two types of encryption: Symmetric & asymmetric encryption.

- **Symmetric encryption** - In symmetric encryption the same key is used for encryption & decryption. It is therefore critical that a secure method is considered to transfer the key between sender & recipient.
- **Asymmetric encryption** - Asymmetric encryption uses the notion of a key pair: a different key is used for the encryption & decryption process. One of the keys is typically known as the private key & the other is known as the public key.

Need of Diffe-Hellman key Exchange algorithms:-

The Diffe-Hellman (DH) Algorithm is a key exchange protocol that enables two parties communicating over public channel to establish a mutual secret without is being transmitted over the internet.

SAMAR

The key exchange protocol is considered an important part of cryptographic mechanism to protect secure end-to-end communications.

Implementation of Diffe-Hellman key exchange algorithms:
- The Diffe-Hellman key exchange algorithm starts with the selection of a prime number plus one of it's primitive roots. (primitive generations)
- What is a primitive root ?

⇒ If the prime number is p, a primitive root (or generator) g is a number, that when n goes from 1 to p-1, then $g^n \bmod p$ goes through all the numbers 1---(p-1) in some order.

- $g^n \bmod p$ means the remainder when you raise g to n & divide by p.

Example :-

$g = 3, p = 5$

$\therefore n = p - 1 = 4$

n	g^n 3^n	$g^n \bmod p$ $3^n \bmod 5$
1	3	3
2	9	4
3	27	2
4	81	1

$\therefore g^n \bmod p$ goes through all the no. 1---4. in 3, 4, 2, 1 order

$\therefore 3$ is a primitive root of 5.

Implementation [Algorithm] :-
- P & G are both publicly available numbers -
- P is at least 512 bits.

SAMAR

- Users pick private values a & b.
- Compute public values
  $x = g^a \bmod p$
  $y = g^b \bmod p$.
- public values x & y are exchanged.
- Compute shared, private key.
  $k_a = y^a \bmod p$

  $k_b = x^b \bmod p$.

Mathematically it can be shown that $k_a = k_b$

- Users now have a symmetric secret key to encrypt.

Example :-

Step 1 - Publicly shared information

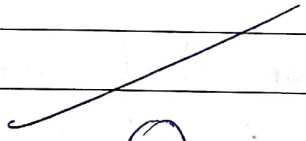
- Alice & Bob publicly agree to a large prime number called the modulus, or p.
- Alice & Bob publicly agree to a number called to generator, or g, which has a primitive root relationship with p.
- Alice & Bob get public numbers.
  $p = 23$, $g = 9$
- Alice & Bob compute public values $a = 4$, $b = 3$
  $x = 9^4 \bmod 23 = 6561 \bmod 23 = 6$
  $y = 9^3 \bmod 23 = 729 \bmod 23 = 16$
- Alice & Bob exchange public numbers
- Alice & Bob compute symmetric keys.
  $k_a = y^a \bmod p = 16^4 \bmod 23 = 9$
  $k_b = x^b \bmod p = 6^3 \bmod 23 = 9$
  $k_a = k_b$
- Now Alice & Bob can talk securely!

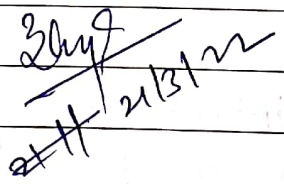
SAMAR

Date / /20

* Conclusion :- In this assignment we have learnt how to implement Diffe-Hellman key exchange algorithm.

* Reference :-
  - Class notes.





ICS

Roll no:- (46)

* Assignment No.:- 02 *

* **Aim:** Implementation of RSA Encryption Algorithm.

* **Input:** Two prime no. → P & q
  k - constant value
  M - Message

* **Output:** - Encrypted data
  - Decrypted data.

* **Theory:-**

- **Public Key Cryptography:-**
  - Public key cryptography also known as Assymmetric key Cryptography.
  - In this two different keys are used, one for encryption & one for decryption.
  - Involves the use of two keys:
    (a) Public key - which may be known by anybody & can be used to encrypt message, & verify signatures.
    (b) Private key - known only to the recipient, used to decrypt message & sign (create) Signature.
  - It is assymmetric because, those who encrypt message or verify signatures cannot decrypt messages or create signatures.

- **RSA Algorithm:-**
  - The RSA algorithm - named after Ron Rivest, Adi Shamir & Leonard Adleman.
  - It is most widely accepted & general - purpose approach to public key encryption.
  - It is based on a property of positive integers, various symbols used in RSA.

(2)

n = a modulus for modular Arithmetic
$\phi(n)$ = the totient of n.
e = an integer that is relatively prime to $\phi(n)$
d = an integer that is the multiplicative inverse of a e modulo $\phi(n)$.
- RSA is a public-key encryption algorithm with a public key of $P_U = \{e, n\}$ & a private key of $P_R = \{d, n\}$.

- Computational steps for key Generation :-
(i) Generates two different primes p & q.
(ii) Calculate the modulus $n = p \times q$
(iii) Calculate the totient $\phi(n) = (p-1) \times (q-1)$
(iv) Select for public exponent an integer 'e' such that $1 < e < \phi(n)$ & $\gcd(\phi(n), e) = 1$.
(v) Calculate for the private exponent a value for d such that $d = e^{-1} \bmod \phi(n)$.
(vi) Public key = [e, n]
(vii) private key = [d, n].

* Conclusion :-
In this assignment, I learnt how to implement RSA algorithm for generating public key & private key.

* Reference :-
- Class notes.

(c)
2019
21/3/22

ICS

Roll no. 46

* Assignment No.:- 03 *

* **Aim:-** Implementation of S-DES algorithm.
* **Input:-** - 8 bit block of plain text
  - 10 bit key
* **Output:-** 8 bit block of ciphertext.
* **Theory:-**
  - **S-DES algorithm:-**

- The Data Encryption standard is a symmetric-key algorithm for the encryption of digital data.
- The S-DES (simplified) encryption algorithm takes
  - an 8-bit block of plaintext (e.g. 10111101)
  - 10-bit key as input.
  - produces an 8-bit block of ciphertext as output.
- The S-DES decryption algorithm takes
  - an 8-bit block of ciphertext
  - the same 10-bit key used.
  - produces the original 8-bit block of plaintext.

* **Algorithm:-**

(i) An initial permutation (IP)
(ii) A complex function labeled f_K, which involves both permutation & substitution operations & depends on key input.
(iii) A simple permutation function that switches (SW) the two halves of the data.
(iv) The function f_K again, and
(v) a permutation function that is the inverse of the initial permutation (IP^{-1}).

Ciphertext = $IP^{-1}(fk_2(SW(fk_1(IP(plaintext)))))$

Where,

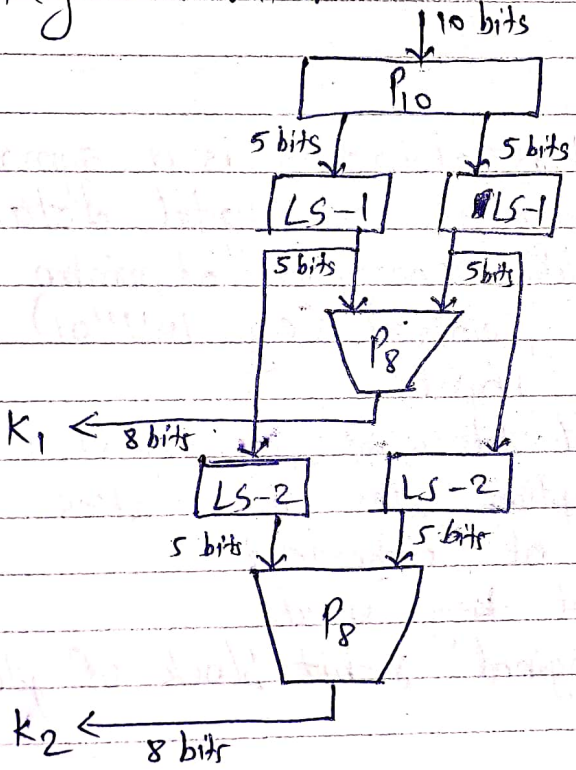
$K_1 = P_8(shift(P_{10}(key)))$

$K_2 = P_8(shift(shift(P_{10}(key))))$

Decryption is essentially the reverse of encryption.

plaintext = $IP^{-1}(fk_1(SW(fk_2(IP(ciphertext)))))$

- Key Generation :-



Permutation
Split key, circular shift

Permutation - K_1 output

split key, circular shift

permutation, K_2 output

★ Conclusion :-

In this assignment, I have learnt how to implement S-DES algorithm.

★ Reference :-

- Class Notes.

© Shuyi 21/3/22

ICS

Assignment - 04

- Aim :- Implementation of S-AES Algorithm. (Simplified Advanced Encryption Standard.)
- Input :-
  - 128 bits key
- Output :-
  - 128 bits of encrypted cipher text.
- Theory :-

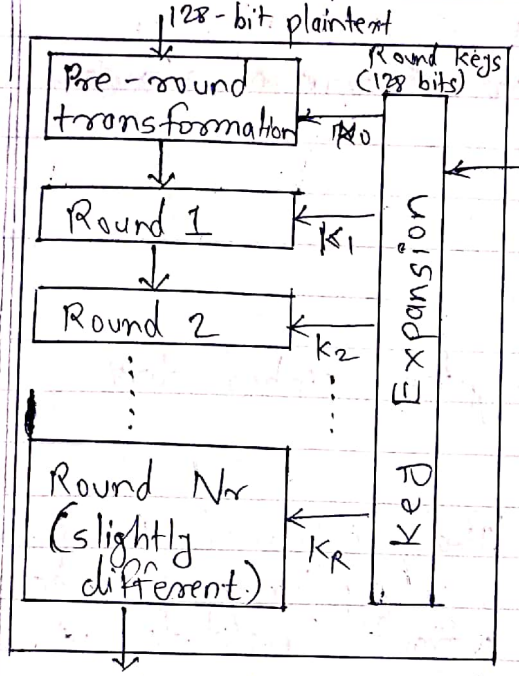
S-AES algorithm :- Simplified Advanced Encryption standard is a specification for the encryption of electronic data established by the U.S. National Institute of standards and Technology (NIST) in 2001.

- AES is widely used today as it is a much stronger than DES & triple DES despite being harder to implement.
- AES is a block cipher.
- The key size can be 128/192/256 bits.
- Encrypts data in blocks of 128 bits each.
- The number of rounds depends on the key length as follows :
  - 128 bit key - 10 rounds
  - 192 bit key - 12 rounds
  - 256 bit key - 14 rounds

4 steps of AES algorithm :-

- byte substitution
- Shift rows
- mix columns
- round key.

- Schematic of AES Structure :-

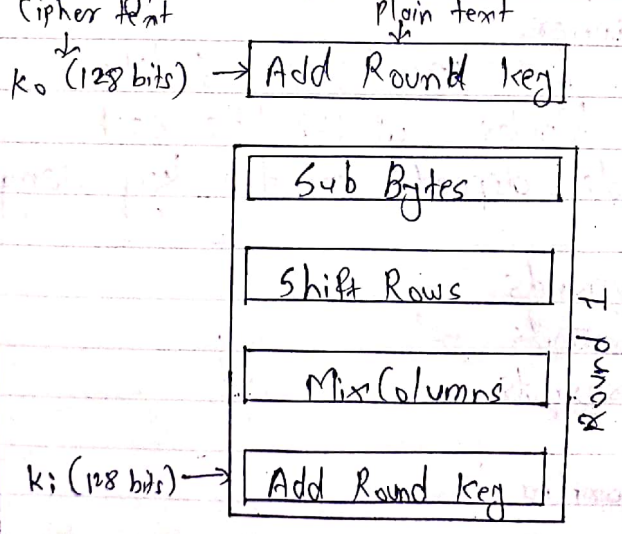


Cipher key (128, 192, or 256 bits)

R	Key Size
10	128
12	192
14	256

Relationship between number of rounds (R) and cipher key size.

- Encryption Process :-



- Conclusion :- I have learned how to implement S-AES algorithm.