11. Keselamatan

11.0 Pertimbangan Keselamatan
11.1 Penyulitan
11.2 Pengenalan dan Kuasa
11.3 Daftar teragih

- Tugasan 18

Tugasan 18

1. Dalam penyulitan tak simetri RSA, nilai-nilai berikut diberi: kekunci rahsia d=11, dan kekunci awam (n=15, e=3).
   Sulitkan mesej m=13 menggunakan kekunci awam, dan kemudian nyahsulitkan menggunakan kekunci rahsia.

   Jika m ini merupakan kekunci rahsia yang dihantar bagi suatu sistem penyulitan bersimetri menggunakan XOR, kodkan (guna Python) mesej berikut (ulang kekunci untuk setiap 4 bit):

   > Pucuk pauh delima batu; air sembilan di tapak tangan; Sungguh jauh negeri yang satu; di mata hilang di hati jangan.

   Kemudian, nyahkodkan teks sifer itu.

2. Cipta suatu fungsi cincang untuk memberikan kod cincang 4 digit integer untuk nama pelajar-pelajar kelas ini. Guna Python untuk menulis aturcara untuk melaksanakannya.

   Berikan kod cincang untuk nama-nama ini. Adakah wujud perlanggaran?

3. Bina suatu rangkaian blok yang cepat dan mudah dengan Python:

   Blok pertama (blok genesis) –

   ```
   transaksi_1="Dompet#1 bayar 1 bitcoin kpd Dompet#2"
   blok_genesis=(transaksi_1)
   print(blok_genesis)
   ```

   Sediakan cincangan –

   ```
   import hashlib # perpustakaan utk cincangan
   def bit_hash(data):
        rtt_data=str(data) # rentetan
        format_data= rtt_data.encode('utf-8') # kodkan
        bit_ccg = hashlib.sha256(format_data).hexdigest()
        # cincang guna algoritma SHA256
        return bit_ccg # kembalikan cincangan
   ```

   Tambahkan rantaian kpd blok – [cincang blok sebelum, data, cincang blok] –

   ```
   # cincang blok genesis
   blok_genesis_ccg = bit_hash((0,transaction_1))
   print(blok_genesis_ccg)
   # buat semula blok genesis, dgn cincang
   blok_genesis=(0,transaction_1,blok_genesis_ccg)
   print(blok_genesis)
   ```

   Buat blok seterusnya –

   ```
   # buat transaksi utk blok 2
   transaksi_2 = "Dompet#1 bayar 2 bitcoin kpd Dompet#2"
   # cincang blok 2
   blok_2_ccg = bit_hash((blok_genesis_ccg, transaksi_2))
   # bina blok 2
   blok_2 = (blok_genesis_ccg,transaksi_2,blok_2_ccg)
   print(blok_2)
   ```

Berikan keluaran aturcara anda.

Fungsi utk buat blok baharu –

```
# cipta fungsi yg membuat suatu blok baharu
# masukkan nombor blok (“ketinggian” blok) sbg
#                  kandungan blok juga
def cipta_blok(seb_blok_ccg,seb_blok_nr,data_transaksi):
    blok_nr = seb_blok_nr + 1
    blok_ccg = bit_hash((seb_blok_ccg,blok_nr,data_transaksi))
    baru_blok= (seb_blok_ccg,blok_nr,data_transaksi,blok_ccg)
    return baru_blok
```

Uji fungsi ini (mula blok nombor 0) –

```
# cetak blok genesis
blok_genesis = cipta_block(0,-1,"Dmp#1 bayar 1 bc kpd Dmp#2")
           # nombor blok bermula pd sifar
print(blok_genesis)
# cetak 3 blok lagi
seb_blok=blok_genesis
for i in range(3):
    seb_blok_ccg=seb_blok[3]
                 # unsur ke3 blok sebelumnya
    seb_blok_nr=seb_blok[1]
                 # unsur ke1 blok sebelumnya
   str_blok=cipta_blok(seb_blok_ccg,seb_block_nr,"Dmp#1 bayar " +
   str(i+2)+" bc kpd Dmp#2")
                 # bayar 1 lebih bc setiap blok
    print(str_blok)
    seb_blok=str_blok
```

Semak bahawa cincang-cincang merantai.