TUGAS 1

PRAKTIKUM KRIPTOGRAFI

Disusun oleh:

Muhamad Rumi Rifai - 140810220026

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

JATINANGOR

2024

Soal

1. Cari penjelasan dan perbedaan lebih lanjut terkait perbedaan kriptografi Klasik dan Modern, termasuk jenis/tipe dari masing-masing kriptografi!
2. Cari 3 (atau lebih) contoh lain dari aplikasi/penerapan kriptografi, dan jelaskan secara singkat peran dan logika kriptografi dalam penerapan tersebut! (Selain yang sudah disebutkan di slide sebelumnya)
3. Cari contoh algoritma kriptografi klasik dan 1 contoh algoritma modern, lalu berikan penjelasan singkatnya!
4. Pilih algoritma dari slide 17, coba eksplorasi terkait algoritma tersebut, lalu tuliskan penjelasan hasil eksplorasi kalian!

Jawaban

1. Ada beberapa perbedaan mendasar antara kriptografi klasik dengan kriptografi modern:
    - Klasik

        Pada kriptografi klasik, metode pengamanan data yang digunakan berbasis pada transformasi sederhana yang melibatkan karakter atau huruf dari suatu pesan, tanpa memanfaatkan kemajuan komputer modern. Fungsi dari pemahaman akan kriptografi klasik adalah untuk pemahaman konsep dasar dari kriptografi itu sendiri.

        Beberapa karakteristik dari kriptografi klasik antara lain menggunakan operasi yang lebih sederhana, tidak melibatkan kunci yang kompleks, serta keamanan yang bergantung pada kerahasiaan algoritma.

        Jenis-jenis dari kriptografi klasik diantaranya seperti subtitusi sandi, transposisi cipher, serta sistem enigma. Beberapa kelemahan dari kriptografi klasik diantaranya mudah dipatahkan dengan analisis frekuensi tertentu, serta algoritmanya yang sederhana rentan terhadap brute-force dan serangan berdasarkan statistik.
    - Modern

        Pada kriptografi modern, perkembangan komputer dan matematika yang lebih canggih membantu dalam proses pengembangan algoritma yang lebih kompleks yang melibatkan operasi matematis, terutama dalam kriptografi kunci publik dan kunci simetris. Dengan terlibatnya komputer, daya yang diproses melibatkan biner 0 dan 1 sehingga segala bentuk proses data digital bekerja dalam mode bit.

        Beberapa karakteristik dari kriptografi modern antara lain .algoritma yang lebih kompleks, menggunakan kunci yang kompleks, lebih aman terhadap serangan brute-force.

        Jenis-jenis dari kriptografi modern diantaranya seperti kriptografi kunci simetris, kriptografi kunci asimetris (publik), dan kriptografi hash, kriptografi quantum (pendekatan masa depan).

Beberapa kekurangan dari kriptografi modern antara lain kompleksitas yang sangat tinggi, serta potensi akan resiko dari komputer quantum.

2. Beberapa contoh lain dari aplikasi/penerapan kriptografi antara lain:
    - Protokol SSL digunakan untuk melakukan browsing secara aman bertindak sebagai protokol yang mengamankan komunikasi client dan server, bekerja dengan memberikan port khusus untuk menerima dan mengirim informasi dari jaringan dalam mode byte stream, lalu diterjemahkan dalam protocol TCP/IP
    - Kriptografi pada blockchain yaitu, diberikan kunci publik (alamat) dan private (transaksi) kepada setiap pengamat arus, lalu menggunakan protokol tanda tangan saat melakukan transaksi, Proof of works(PoW) dan Proof of Strike (PoS), dan merkle tree
    - BitLocker melindungi data pada hard disk dimiliki oleh OS seperti windows dan Mac OS dengan cara mengenkripsi semua disk jika terjadi hal yang tidak diinginkan.
3. Beberapa contoh algoritma kriptografi:
    - Klasik : Caesar Cipher
      Caesar Cipher adalah salah satu algoritma kriptografi klasik paling sederhana. Diciptakan oleh Julius Caesar, algoritma ini bekerja dengan cara menggantikan setiap huruf dalam teks asli dengan huruf lain yang berjarak tetap di sepanjang alfabet.
      Cara Kerja:
        - Setiap huruf dalam teks dienkripsi dengan menggeser posisinya dalam alfabet dengan jumlah tetap. Misalnya, jika pergeserannya adalah 3, huruf A akan digantikan oleh D, B oleh E, dan seterusnya.
        - Contoh: Jika kita ingin mengenkripsi kata "HELLO" dengan pergeseran 3:
            - H -> K
            - E -> H
            - L -> O
            - O -> R
        - Hasilnya: "HELLO" menjadi "KHOOR".

      Kelebihan dari algoritma ini adalah mudah diimplementasikan dan dipahami. Sedangkan kelemahan adalah sangat rentan terhadap serangan analisis frekuensi dan brute force, karena hanya ada 25 kemungkinan pergeseran.
    - Modern : RSA (Rivest-Shamir-Adleman)
      RSA adalah algoritma kriptografi modern yang menggunakan prinsip matematika dari teori bilangan dan faktor bilangan prima besar. RSA adalah salah satu algoritma kunci publik yang paling banyak digunakan dalam komunikasi aman.
      Cara Kerja:
        - RSA menggunakan dua kunci: kunci publik dan kunci privat.
            - Kunci publik digunakan untuk mengenkripsi pesan.

    - Kunci privat digunakan untuk mendekripsi pesan.
  - Algoritma ini bergantung pada kesulitan memfaktorkan bilangan bulat besar, sehingga meskipun kunci publik dapat diketahui oleh siapa saja, kunci privat tetap aman.

  Langkah Dasar:

  1. Pilih dua bilangan prima besar (misalnya p dan q).
  2. Hitung $n = p \times q$ dan $\varphi(n) = (p-1) \times (q-1)$.
  3. Pilih kunci publik e yang relatif prima terhadap $\varphi(n)$.
  4. Hitung kunci privat d sedemikian rupa sehingga $(e \times d) \equiv 1 \pmod{\varphi(n)}$.
  5. Untuk enkripsi, lakukan operasi: ciphertext = (plaintext^e) mod n.
  6. Untuk dekripsi, lakukan operasi: plaintext = (ciphertext^d) mod n.

  Contoh: Jika Alice ingin mengirim pesan ke Bob, dia menggunakan kunci publik Bob untuk mengenkripsi pesan. Hanya Bob yang dapat mendekripsinya dengan kunci privatnya.

  Kelebihan dari algoritma ini adalah sangat aman, asalkan ukuran kunci cukup besar serta digunakan dalam banyak protokol keamanan modern seperti HTTPS. Sedangkan kelemahan dari algoritma ini adalah relatif lambat dibandingkan dengan algoritma simetris dan ukuran kunci yang besar diperlukan untuk tingkat keamanan yang lebih tinggi.

4. Salah satu algoritma hashing satu arah. Nama lainnya adalah SHA-0, disempurnakan menjadi SHA-1, keluaran terbaru SHA-2 dan SHA3. Tidak digunakan karena memiliki kelemahan dalam hal keamanan karena ditemukan kerentanan terhadap serangan tabrakan (collision attack). Ini berarti dua pesan yang berbeda bisa menghasilkan hash yang sama, yang seharusnya tidak mungkin dalam algoritma hash yang aman. Kelemahan ini menyebabkan SHA-0 digantikan oleh SHA-1 yang memiliki modifikasi untuk meningkatkan keamanan.

   Langkah algoritma :

   1. **Padding (Penambahan Padding)**
      - **Tujuan:** Memastikan bahwa panjang pesan menjadi kelipatan dari 512 bit.
      - **Proses:**
        1. Tambahkan bit 1 di akhir pesan asli.
        2. Tambahkan bit 0 sebanyak yang diperlukan hingga panjang pesan mencapai 448 bit modulo 512 (berarti pesan tersebut 64 bit lebih pendek dari kelipatan 512 bit).
        3. Tambahkan representasi biner 64-bit dari panjang asli pesan sebelum padding. Ini membuat panjang total pesan menjadi kelipatan 512 bit.
   2. **Pembagian Pesan menjadi Blok 512-bit**
      - **Proses:** Setelah padding, pesan dibagi menjadi blok-blok 512-bit. Setiap blok 512-bit kemudian dipecah menjadi 16 kata 32-bit.

3. **Inisialisasi Variabel Hash**
    - **Tujuan**: Menetapkan nilai awal untuk lima variabel hash internal.
    - **Nilai Awal:**
        - A = 0x67452301
        - B = 0xEFCDAB89
        - C = 0x98BADCFE
        - D = 0x10325476
        - E = 0xC3D2E1F0
4. **Pemrosesan Setiap Blok 512-bit**
    - **Proses:** Untuk setiap blok 512-bit, SHA-0 melakukan sejumlah operasi bitwise dan matematika dalam 80 langkah. Langkah-langkah ini termasuk:
        1. **Ekspansi Pesan:**
            - Buat 80 kata 32-bit Wt dari 16 kata awal 32-bit. Kata-kata tambahan ini dihitung sebagai:

$$W_t = W_{t-16} \oplus W_{t-14} \oplus W_{t-8} \oplus W_{t-3}$$

            - Di mana t adalah indeks kata yang sedang dihitung, dan $\oplus$ adalah operasi XOR.
        2. **Iterasi Melalui 80 Langkah**:
            - Untuk setiap langkah t dari 0 hingga 79, lakukan operasi berikut:

$$TEMP = (A \ll 5) + f(B,C,D) + E + W_t + K_t$$

        3. **Fungsi-fungsi Logika**:
            - Tiga fungsi logika yang digunakan dalam SHA-0:
                - Untuk $0 \leq t \leq 19$

$$f(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$

                - Untuk $20 \leq t \leq 39$

$$f(B,C,D) = B \oplus C \oplus D$$

                - Untuk $40 \leq t \leq 59$

$$f(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$$

                - Untuk $60 \leq t \leq 79$

$$f(B,C,D) = B \oplus C \oplus D$$

5. **Pembaharuan Nilai Hash**
    - Setelah 80 langkah untuk setiap blok 512-bit selesai, nilai hash internal diperbarui dengan menambahkan nilai hasil dari langkah-langkah di atas ke nilai hash sebelumnya.
6. **Hasil Akhir (Message Digest)**
    - Setelah semua blok 512-bit diproses, nilai hash akhir dari A,B,C,D,EA, B, C, D, EA,B,C,D,E digabungkan untuk membentuk output 160-bit, yang merupakan hasil dari algoritma SHA-0.