Ovo je još jedan klasičan zadatak iz kriptografije koji kombinuje heširanje (otisak) i automatizaciju putem Bash skriptovanja.

Evo detaljnog pojašnjenja zadatka i skripte.

🧐 Pojašnjenje Zadatka 4

1. Cilj Zadatka

Glavni cilj (prema **prvoj slici**) je **odrediti parove:**

1. **Ulazna datoteka** (iz liste ulaz1.txt do ulaz50.txt).
2. Algoritam za heširanje.

   ... koji su korišćeni za generisanje svakog pojedinog otiska (iz datoteke otisci.dec).

2. Dati Materijali

- **Ulazne datoteke:** Veliki broj datoteka (ulaz1.txt do ulaz50.txt) koje sadrže potencijalne ulazne podatke (lozinke, ključeve, fraze) za heširanje.
- **Datoteka sa otiscima:** Datoteka otisci.dec koja sadrži listu heševa (otisaka) koje treba probiti.

3. Kontekst Heširanja

Ovaj zadatak se oslanja na činjenicu da openssl passwd može generisati heševe koristeći **različite algoritme** (koji su interno numerisani ili nazvani). Zadatak zahtijeva da se pronađe ne samo koji je ulazni tekst korišćen, već i **koji algoritam** je primijenjen da se dobije određeni heš.

💻 Pojašnjenje Skripte (skripta.sh)

Skripta (sa **druge slike**) je dizajnirana da automatski provjerava sve moguće kombinacije ulaznih datoteka i pretpostavljenih algoritama, pokušavajući da se podudari sa ciljnim otiscima iz otisci.dec.

Analiza Skripte Liniju po Liniju:

1. Učitavanje Ulaznih Podataka i Otpaka

Bash

```
otisci=$(cat otisci.dec)
ulazi="ulaz*"
```

- otisci: Učitava **cijeli sadržaj** datoteke otisci.dec (listu ciljnih heševa, od kojih svaki vjerovatno sadrži i informaciju o algoritmu i saltu) u jednu varijablu.
- ulazi: Džoker (ulaz*) koji će Bash koristiti za listanje svih datoteka koje počinju sa ulaz (npr. ulaz1.txt do ulaz50.txt).

2. Vanjska Petlja (Iteracija kroz Otiske)

Bash

```
for otisak in $otisci
do
   # ...
done
```

- Petlja prolazi kroz svaki red (svaki pojedinačni heš/otisak) iz datoteke otisci.dec.

3. Parsiranje (Odvajanje) Algoritma i Salta

```
Bash
   algoritam=$(echo "$otisak" | cut -d '$' -f 2)
   salt=$(echo "$otisak" | cut -d '$' -f 3)
```

- Ovo je **ključni dio**. Mnogo formata heširanja (npr. MD5, SHA-512) u OpenSSL-u i Linux-u koristi strukturu poput $...algoritam$...salt$...heš....
- cut -d '$': Koristi znak $ kao delimiter (separator).
- -f 2: Uzima drugu kolonu/polje (koja obično sadrži **algoritam**).
- -f 3: Uzima treću kolonu/polje (koja obično sadrži **salt**).
- **Zaključak:** Skripta pretpostavlja da je format svakog otiska u otisci.dec takav da se algoritam i salt mogu izvući na ovaj način.

4. Unutrašnja Petlja (Iteracija kroz Potencijalne Ulaze)

```
Bash
for ulaz in $ulazi
do
   # ...
done
```

- Ova petlja prolazi kroz sve datoteke ulazX.txt.

5. Pokušaj Heširanja i Provjera

```
Bash
   sadrzaj=$(cat "$ulaz")
   otisak2=$(openssl passwd -"$algoritam" -salt "$salt" "$sadrzaj" 2>error1.txt)

   if [[ "$otisak" == "$otisak2" ]]
   then
      # Pronadjeno podudaranje!
      # ...
   fi
```

- sadrzaj: Učitava sadržaj trenutne datoteke (ulazX.txt).
- otisak2: Ovo je **probnih heš**.
  - openssl passwd: Generiše heš lozinke.
  - -"$algoritam": Koristi algoritam izvučen iz ciljnog otiska (npr. -apr1, -5, -6).
  - -salt "$salt": Koristi salt izvučen iz ciljnog otiska.
  - "$sadrzaj": Hešira se sadržaj iz ulazX.txt.
- if [["$otisak" == "$otisak2"]]: **Poređenje.** Ako je probni heš (otisak2) identičan ciljnom hešu (otisak), pronašli smo ispravan par: (Ulazna datoteka, Algoritam).
- break: Prekida unutrašnju petlju jer je za trenutni otisak pronađen odgovarajući ulaz.

6. Rješavanje Zadatka Ovom Skriptom

Skripta rješava zadatak putem **brute-force** (grube sile) heširanja, ali sa preciznošću:

1. Za **svaki ciljni otisak** u otisci.dec.
2. Pretpostavlja da su **algoritam i salt** već ugrađeni u format tog otiska i parsira ih.

3. Pokušava da hešira **svih 50 ulaznih datoteka** koristeći taj algoritam i salt.
4. Kada se heš **poklopi**, ispisuje rezultat (koji je ulaz i koji je algoritam) i nastavlja sa sledećim ciljnim otiskom.

Rezultati (Pretpostavljeni iz komentara):

Komentarisane linije (#Datoteka: ulaz14.txt, #Algoritam: 1, itd.) sugerišu da bi rezultat skripte trebao biti:

- Jedan otisak odgovara **ulaz14.txt** sa **algoritmom 1** (MD5).
- Jedan otisak odgovara **ulaz22.txt** sa **algoritmom 5** (SHA-256).
- Jedan otisak odgovara **ulaz40.txt** sa **algoritmom apr1** (Apache MD5).
- Jedan otisak odgovara **ulaz41.txt** sa **algoritmom 6** (SHA-512).

☑ Sugestije

1. **Provjera Parsiranja:** Iako skripta koristi `cut` za izdvajanje, to je ispravno samo ako svi otisci u `otisci.dec` prate isti format. U formatima heševa OpenSSL/Linux:
   - `$1$`: MD5 (algoritam `1` je često implicitni dio `$1$`)
   - `$apr1$`: Apache MD5 (algoritam `apr1`)
   - `$5$`: SHA-256 (algoritam `5`)
   - `$6$`: SHA-512 (algoritam `6`)

   Skripta je napisana da izvuče taj broj/naziv (npr. `1`, `apr1`, `5`, `6`) i koristi ga.
2. **Bolje Upravljanje Izlazom:** Skripta trenutno ispisuje samo pronađene rezultate. Kao poboljšanje, možete proslediti izlaz u datoteku za lako snimanje rješenja:
   Bash
   ```
   # Unutar if bloka:
   echo "Datoteka: $ulaz, Algoritam: $algoritam" >> pronadjeni_parovi.txt
   ```

Da li biste željeli da pogledamo tačne komande za dešifrovanje iz prvog zadatka, sada kada smo analizirali i mehanizam traženja ključa i mehanizam traženja ulaza?