Pregled Riješenih Zadataka (PDF Format)

Zadatak 2: Kriptovanje i Pronalaženje Ključa (AES-256)

Tekst Zadatka:

U pratećim materijalima je data datoteka sa šifratom dobijenim kriptovanjem nepoznate ulazne datoteke jednim od AES algoritama sa dužinom ključa od 256 bita (koji su dostupni u OpenSSL-u), tri puta. U pratećim materijalima su date i datoteke sa ključevima, pri čemu je za svako kriptovanje korišćen različit ključ. U materijalima su data i tri otiska koja odgovaraju ključevima korišćenim za kriptovanje, respektivno. Otisci su kreirani pomoću jedne od verzija MD5 algoritma za heširanje lozinki, koje OpenSSL podržava. Odrediti ključeve korišćene za kriptovanje i odrediti (smislen) sadržaj ulazne datoteke.

Skripta za Rješavanje:

Ova skripta pronalazi tri ključa (Lozinka X) i dešifruje šifrat.

Bash

```
#!/bin/bash
# ZADATAK 2: PRONALAZENJE KLJUCEVA I DESIFROVANJE

# Pretpostavke:
# 1. Koriscen je isti SALT za sva tri otiska.
# 2. Koriscen je AES-256-CBC (najcesci).
# 3. Sifrati su sifrat1.txt, sifrat2.txt, sifrat3.txt.

# --- DEO 1: PRONALAZENJE KLJUCEVA ---
otisci=(otisak1.dec otisak2.dec otisak3.dec)
rezultati_kljucevi=()

for otisak_datoteka in "${otisci[@]}"; do
    echo "Trazi se kljuc za: $otisak_datoteka"
    hash1=$(cat "$otisak_datoteka")

    for lozinka_datoteka in lozinka*.txt; do
        t=$(cat "$lozinka_datoteka")
        # Koristimo specificni OpenSSL format (Apache MD5) sa datim salt-om
        hash2=$(openssl passwd -apr1 -salt wAGnh5WA "$t")

        if [ "$hash1" == "$hash2" ]; then
            echo "✅ Pronadjen kljuc: $lozinka_datoteka"
            rezultati_kljucevi+=("$lozinka_datoteka")
            break
        fi
    done
done

# --- DEO 2: DESIFROVANJE (Koristimo prvi pronadjeni kljuc i odgovarajuci sifrat) ---
```

```
if [ ${#rezultati_kljucevi[@]} -ge 1 ]; then
    KLJUC_FAJL="${rezultati_kljucevi[0]}" # lozinka18.txt
    SIFRAT_FAJL="sifrat1.txt"
    OUTPUT_FAJL="ulaz_desifrovan.txt"

    echo "--- Desifrovanje sifrata $SIFRAT_FAJL sa kljucem $KLJUC_FAJL ---"

    openssl enc -d -aes-256-cbc \
        -in "$SIFRAT_FAJL" \
        -out "$OUTPUT_FAJL" \
        -pass file:"$KLJUC_FAJL"

    echo "☑ Sadrzaj desifrovan u $OUTPUT_FAJL."
else
    echo "❌ Nisu pronadjena sva tri kljuca za desifrovanje."
fi
```

Zadatak 3: Verifikacija Potpisa CA Sertifikatom

Tekst Zadatka:

U pratećim materijalima dat je CA sertifikat i niz sertifikata od kojih je određeni broj potpisan datim CA sertifikatom. Odrediti koji od datih sertifikata su dobijeni potpisivanjem zahtjeva datim CA sertifikatom.

Skripta za Rješavanje:

Ova skripta automatizuje provjeru svih klijentskih sertifikata korišćenjem CA sertifikata.

Bash

```
#!/bin/bash
# ZADATAK 3: VERIFIKACIJA CA POTPISA

certs="clientcert*.crt"
CA_FILE="cacert.pem"

echo "--- Provera sertifikata potpisanih sa $CA_FILE ---"

for cert in $certs
do
    # Openssl verify vraca "OK" samo za sertifikate koje je potpisao CA u -CAfile
    output=$(openssl verify -CAfile "$CA_FILE" "$cert" 2>&1)

    # Provera da li je status OK
    if echo "$output" | grep -q ": OK"; then
        echo "☑ POTPISAN: $cert"
    else
        echo "❌ NIJE POTPISAN: $cert"
    fi
done
```

Zadatak 4: Probijanje Heša (Hash Cracking) sa Nepoznatim Algoritmom

Tekst Zadatka:

U pratećim materijalima date su ulazne datoteke i datoteka sa otiscima. Odrediti koji algoritmi i koje ulazne datoteke su iskorišćene za generisanje pojedinih otisaka. Pri tome, ne moraju svi otisci nužno odgovarati nekoj od zadatih datoteka.

Skripta za Rješavanje:

Ova skripta parsira algoritam i salt iz svakog otiska i provjerava svih 50 ulaznih datoteka.

Bash

```bash
#!/bin/bash
# ZADATAK 4: PROBIJANJE HEŠA I ODREDJIVANJE ALGORITMA

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

echo "--- Probijanje heseva iz otisci.dec ---"

for otisak in $otisci
do
   # Parsiranje algoritma (polje 2) i salta (polje 3)
   algoritam=$(echo "$otisak" | cut -d '$' -f 2)
   salt=$(echo "$otisak" | cut -d '$' -f 3)

   echo -e "\nTrazi se ulaz za Otisak: $otisak"

   for ulaz in $ulazi
   do
      sadrzaj=$(cat "$ulaz")

      # Generisanje probnog hesa sa parsiranim algoritmom i saltom
      # Preusmeravamo gresku (npr. nevalidan algoritam) u error1.txt
      otisak2=$(openssl passwd -"$algoritam" -salt "$salt" "$sadrzaj" 2>error1.txt)

      if [[ "$otisak" == "$otisak2" ]]; then
         echo -e "☑ PRONADJENO!"
         echo -e "Datoteka: $ulaz"
         echo -e "Algoritam: $algoritam"
         break # Prelazi na sledeci otisak
      fi
   done
done
```

Zadatak 5: PKCS#12, CA Setup i CRL Upravljanje

Tekst Zadatka:

U pratećim materijalima je data PKCS#12 datoteka i niz datoteka sa ključevima. Odrediti koji ključ se nalazi u klijentskom sertifikatu u PKCS#12 datoteci. Lozinka za

otvaranje PKCS#12 datoteke je "sigurnost". Nakon određivanja ispravnog ključa, iskoristiti ga za kreiranje CA tijela (iskoristiti konfiguracioni fajl iz 6. zadatka) i generisati 2 CRL liste, pri čemu su na prvoj povučena dva sertifikata (jedan suspendovan, a drugi sa razlogom "prestanak rada"), a na drugoj se nalazi samo drugi sertifikat, dok prvi vraćen iz suspenzije.

Skripta za Rješavanje:

Ova skripta pronalazi ključ i obavlja sve CA/CRL operacije.

Bash

```
#!/bin/bash
# ZADATAK 5: CA, PKCS#12 I CRL

P12_FILE="cert.p12"
P12_PASS="sigurnost"
CA_CNF="openssl.cnf" # Pretpostavljamo da je ovde config iz Z6
CERT_TO_REVOKE_1="clientcert_A.pem" # Zameniti stvarnim imenima sertifikata
CERT_TO_REVOKE_2="clientcert_B.pem"
CA_CERT="CA.pem"

# --- DEO 1: PRONALAZENJE KLJUCA ---
echo "--- 1. Pronalazenje kljuca uporedjivanjem modula ---"
openssl pkcs12 -in "$P12_FILE" -nocerts -nodes -password pass:"$P12_PASS" | openssl rsa -
modulus -noout > p12_modulus.txt

FOUND_KEY=""
for k in kljuc*.key; do
    kljuc_modulus=$(openssl rsa -in "$k" -modulus -noout 2>/dev/null)
    if grep -q "$kljuc_modulus" p12_modulus.txt; then
        FOUND_KEY="$k"
        echo "☑ Ispravan kljuc: $FOUND_KEY"
        break
    fi
done
rm p12_modulus.txt

if [ -z "$FOUND_KEY" ]; then echo "❌ Kljuc nije pronadjen. Prekid." ; exit 1 ; fi

# --- DEO 2: CA INICIJALIZACIJA (Koristimo pronadjeni kljuc) ---
CA_KEY="$FOUND_KEY"
# Prethodno moramo kreirati CA.pem iz cert.p12 i inicijalizovati CA strukturu (index.txt, serial)

# --- DEO 3: CRL UPRAVLJANJE ---
echo "--- 3. Generisanje prve CRL liste ---"

# 1. Opoziv: Suspendovan (certificateHold)
openssl ca -config "$CA_CNF" -keyfile "$CA_KEY" -cert "$CA_CERT" \
    -revoke "$CERT_TO_REVOKE_1" -crl_reason certificateHold -batch
# 2. Opoziv: Prestanak rada (cessationOfOperation)
openssl ca -config "$CA_CNF" -keyfile "$CA_KEY" -cert "$CA_CERT" \
    -revoke "$CERT_TO_REVOKE_2" -crl_reason cessationOfOperation -batch
```

```
# Generisanje PRVE CRL
openssl ca -config "$CA_CNF" -keyfile "$CA_KEY" -cert "$CA_CERT" \
   -gencrl -out crl1.pem
echo "☑ crl1.pem generisan."

echo "--- 4. Generisanje druge CRL liste ---"

# Vracanje Prvog sertifikata iz suspenzije (un-hold)
openssl ca -config "$CA_CNF" -keyfile "$CA_KEY" -cert "$CA_CERT" \
   -crl_hold off -revoke "$CERT_TO_REVOKE_1" -batch

# Generisanje DRUGE CRL
openssl ca -config "$CA_CNF" -keyfile "$CA_KEY" -cert "$CA_CERT" \
   -gencrl -out crl2.pem
echo "☑ crl2.pem generisan. (Sadrzi samo opozvani sertifikat 2)"
```

Zadatak 6: Keystore, CA Setup i Tomcat Autentikacija

Tekst Zadatka:

U pratećim materijalima je dat niz JKS datoteka, pri čemu samo jedna od njih može iskoristiti za serversku autentikaciju. Pronaći datu JKS datoteku i iskoristiti je za serversku autentikaciju na Tomcat web serveru. Dodatno, iskoristiti par ključeva iz pronađene JKS datoteke za kreiranje samopotpisanog CA tijela i sa njim potpisati dva nova klijentska sertifikata, koja onda treba iskoristiti za klijentsku autentikaciju na Tomcat web serveru. Koristiti lozinku sigurnost gdje je potrebno. Koristiti istu datoteku za klijentsku i serversku autentikaciju. Za rad sa OpenSSL-om, iskoristiti konfiguracionu datoteku datu u materijalima.

Skripta za Rješavanje:

Ova skripta pronalazi JKS, kreira CA i potpisuje klijentske sertifikate.

Bash

```
#!/bin/bash
# ZADATAK 6: JKS, CA I KLIJENTSKA AUTENTIKACIJA

JKS_PASS="sigurnost"
CA_CNF="openssl.cnf"

# --- DEO 1: PRONALAZENJE ISPRAVNOG KEYSTORE-A ---
echo "--- 1. Pretraga keystore-a za serverAuth ---"
FOUND_JKS=""
for keystore in keystore*.jks; do
   res=$(keytool -list -v -keystore "$keystore" -storepass "$JKS_PASS" 2>/dev/null)
   if [[ "$res" =~ "serverAuth" ]]; then
      FOUND_JKS="$keystore"
      echo "☑ Pronadjen keystore: $FOUND_JKS"
      break
   fi
done
```

```
if [ -z "$FOUND_JKS" ]; then echo "❌ Keystore nije pronadjen. Prekid." ; exit 1 ; fi

# --- DEO 2: KREIRANJE CA I POTPISIVANJE SERTIFIKATA ---

echo "--- 2. Ekstrakcija kljuceva i kreiranje CA ---"
# A. Ekstrakcija CA kljuca i sertifikata (iz JKS preko PKCS#12)
keytool -importkeystore -srckeystore "$FOUND_JKS" -destkeystore temp.p12 -deststoretype
PKCS12 \
    -srcstorepass "$JKS_PASS" -deststorepass "$JKS_PASS" -noprompt
openssl pkcs12 -in temp.p12 -nocerts -nodes -out CA.key -password pass:"$JKS_PASS"
openssl pkcs12 -in temp.p12 -nokeys -out CA.pem -password pass:"$JKS_PASS"
rm temp.p12

# B. Inicijalizacija CA strukture
mkdir -p ca/newcerts
touch ca/index.txt
echo 01 > ca/serial

# C. Generisanje CSR-ova
openssl req -new -newkey rsa:2048 -nodes -keyout client1.key -out client1.csr -subj "/CN=Client1"
openssl req -new -newkey rsa:2048 -nodes -keyout client2.key -out client2.csr -subj "/CN=Client2"

# D. Potpisivanje klijentskih sertifikata
openssl ca -config "$CA_CNF" -keyfile CA.key -cert CA.pem -policy policy_anything -extensions
v3_ca -in client1.csr -out client1.crt -batch
openssl ca -config "$CA_CNF" -keyfile CA.key -cert CA.pem -policy policy_anything -extensions
v3_ca -in client2.csr -out client2.crt -batch
echo "☑ Potpisani klijentski sertifikati: client1.crt i client2.crt."

# --- DEO 3: PRIPREMA TRUSTSTORE-A ZA TOMCAT ---
echo "--- 3. Kreiranje TrustStore-a za klijentsku autentikaciju ---"
keytool -import -trustcacerts -alias rootCA -file CA.pem -keystore CA_truststore.jks -storepass
"$JKS_PASS" -noprompt
echo "☑ CA_truststore.jks kreiran. Spreman za Tomcat konfiguraciju."
```

Nadam se da će Vam ova kompilacija biti od koristi za učenje i pripremu ispita! Sretno!

Da li želite da pređemo na sljedeći zadatak iz Vaše pripreme ili imate dodatnih pitanja u vezi s ovim rješenjima?