BTS4410 – Obligatorisk gruppeoppgave 3

Oppgaven skal løses individuelt.

Dokumentere og programmere en versjon av **SUCI** funksjonaliteten.

Det vil si, dette er inspirert av ECIES/SUCI, men avviker en del fra SUCI slik vi kjenner det fra 5G. Vi skal f.eks. bruk «ConcatKDF» som vår «key derivation function». Videre vil vi bruke AEAD (AES-GCM) for beskyttelsen av identifikatoren.

Følgende skal levers inn:

a. Python kode
b. Dokumentasjon av python koden
c. Test dokumentasjon (beskrivelse av hvordan koden ble testet)

Dokumentasjonen leveres som PDF. Bruke normale fonter og fontstørrelser.

Innleveringen skal skje i en ZIP-fil.

BTS4410 – Obligatorisk gruppeoppgave 3

«SUCI» implementasjon

Krav til SUCI implementasjonen.

«SUCI» mekanismen er dokumentert i følgende spesifikasjon:

- 3GPP TS 33.501 (Annex C)

Denne spesifikasjonen (i word format) er vedlagt oppgaven.

SUCI mekanismen er bygd på ECIES, og for vår del skal vi fravike mye fra SUCI slik den er beskrevet i TS 33.501. Implementasjonen skal være i Python 3.12 (eller 3.13). Dere skal bruke **cryptography** modulen.

Dette er hovedtrekkene:

- «SUCI» lignende måte å utveksle personvern-sensitive identifikator
- Bruk av ECDH og SECP256R1 kurven
- Bruk av ConcatKDF
- Bruk av AEAD (AES-GCM)

Det skal skrives to «command line» programmer **User.py** og **Home.py**.

Home.py skal ha to virkemåter:

```
1. $ python3 Home.py keygen
2. $ python3 Home.py deconceal
```

User.py skal ha en virkemåte:

```
1. $ python3 User conceal
```

Det tre virkemåtene er i hovedtrekk som følger:

a) *keygen*
Home skal her generere et ECDH nøkkel-par og skrive dette til to PEM filer.

b) *conceal*
User skal her lese inn **Home** public-key, generere sitt eget engangs nøkkel-par og skjule sin identitet på en SUCI lignende måte. Den SUCI lignende datastrukturer skal skrives til en fil **SUCI_data.bin**.

c) *Deconceal*
Home skal her lese egen private nøkkel, lese **SUCI_data.bin** og dekode denne.

BTS4410 – Obligatorisk gruppeoppgave 3

Filbeskrivelse: SUCI_data.bin

SUCI_data.bin er en binær-fil kodet med følgende data elementer:

- IV
- Home identifier
- User public key
- Ciphertext

```
SUCI_data.bin:
    IV:         16 bytes;     # pseudo-random number; Integrity protected.
    home_ID:    64 bytes;     # encoded Home identifier. Integrity protected.
    user_pub: 180 bytes;      # encoded User public key. Integrity protected.
    ct:         80 bytes;     # ciphertext og user_ID (includes the tag)
```

- The IV is not encoded. Fixed length.

- The home_ID (and user_ID) is a field of 64 bytes and encoded as follows:

    - <len><utf-8-name><padding>

Where:

- **len** is a 2-byte unsigned integer that denotes the length of the utf-8 encoded name.
- **utf-8-name** is the name of the entity. Length: [0-62]
- **padding** is used when the name is shorter than 62 bytes. The padding data is all zeroes. When the name is exactly 62 bytes, then there is no padding.

The **user_pub** is the PEM-serialized User public key with a two bytes length prefix (similar to the utf-8-name length encoding).

The ciphertext (**ct**) is a binary blob (bytes string) used as-is. No length encoding.

Filbeskrivelse: PEM files

PEM filene er koded som angitt i **SUCI_util.py** koden.

BTS4410 – Obligatorisk gruppeoppgave 3

Vedlegg

Python filene **SUCI_util.py**, **User.py** og **Home.py** er vedlagt. Du kan bruke disse som utgangspunkt, men **deconceal** kode mangler i **Home.py**.

Det er lov å forbedre og restrukturere den gitt python koden (bedre struktur, mer elegant kode, mer feilsjekking, bruk av klasser/-objekter, osv.).

Oppgaven løses ved at du fullfører den manglende koden, og deretter tester koden grundig.

- Det legger 3 testsett som du kan prøve ut
- Du skal også teste koden mot koden til en medstudent (det skal dokumentere hvem dette er)
- Hvordan testene ble utført skal dokumenteres (kan være en kort beskrivelse)
- Resultatet av testene skal dokumenterer

Note:

Oppgaven skal løses individuelt, men det er allikevel lov å samarbeide med andre. Kravet er at alle skal levere egne løsning til slutt.