БЕЗБЕДНОСТ НА КОМПЈУТЕРСКИ СИСТЕМИ

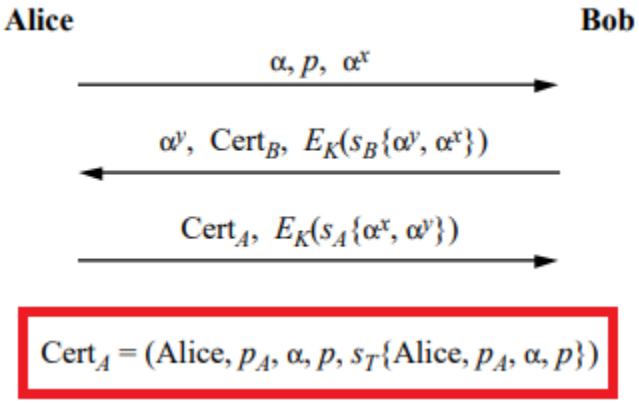
ЛАБОРАТОРИСКА ВЕЖБА 3

Фисник Лимани, 151027

ДОКУМЕНТАЦИЈА

1. Certificate.java

- Оваа класа ни го претставува сертификатот
- Во оваа класа се чуваат овие информации:
  - **String name**
    - Името на засегнатата страна која сака да комуницира
  - **byte[] publicKey**
    - Јавниот клуч
  - **BigInteger alpha**
    - Diffie-Hellman параметарот (примитивен елемент)
  - **BigInteger p**
    - Diffie-Hellman параметарот (прост број)
  - **byte[] signedBytes**
    - Потписот од страна на CA
- Ги чуваме сите информации кој можат да се видат во заокруженият дел со црвено подолу (во шемата на протколот):



2. CertificateRequest.java

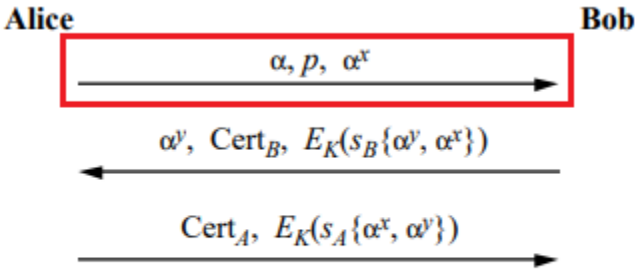
- Оваа класа ни претставува барање до CA за сертификат
- Тука се чуваат овие информации:
  - **String name**
  - **byte[] publicKey**
  - **BigInteger alpha**
  - **BigInteger p**
- Дефинициите ги имаме на класата погоре (тука го имаме истата класа како погоре со разлика на тоа што тука го немаме делот на **signedBytes** која ќе биде додадена од страна на CA-то)

3. CertificateAuthority.java

- Во оваа класа се имплементира CA-то
- Тука чуваме:
  - PrivateKey privateKey
    - Приватниот клуч на CA
  - PublicKey publicKey
    - Јавниот клуч на CA
  - MessageDigest digest
    - Изгенерирање на хешот
  - Map<String, PublicKey> saved
    - Регистрираните корисници (страни) со нивните соодветни јавни клучеви
- Функции:
  - addUser(User u)
    - додадеме (регистрираме) нов корисник со неговиот јавен клуч
  - sign(CertificateRequest cr)
    - издава сертификат/го потпишува барањето за сертификат

4. FirstMessage.java

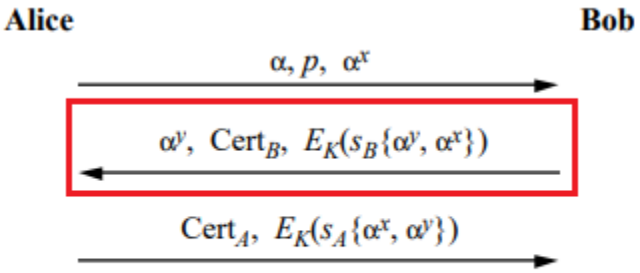
- Ни ја претставува првата порака што се испраќа од Алис кон Боб



$$\text{Cert}_A = (\text{Alice}, p_A, \alpha, p, s_T\{\text{Alice}, p_A, \alpha, p\})$$

5. SecondMessage.java

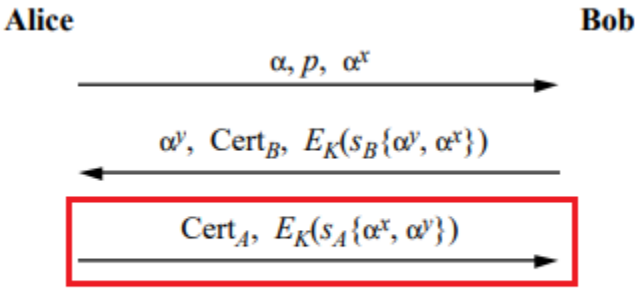
- Ни ја претставува втората порака што се испраќа од страна на Боб до Алис



$$\text{Cert}_A = (\text{Alice}, p_A, \alpha, p, s_T\{\text{Alice}, p_A, \alpha, p\})$$

6. ThirdMessage.java

- Ни го претставува третата порака што се испраќа од страна на Алис до Боб



$$\text{Cert}_A = (\text{Alice}, p_A, \alpha, p, s_T\{\text{Alice}, p_A, \alpha, p\})$$

7. User.java

- Оваа класа ни го претставува корисникот (како на пр. Алис или Боб)
- Тука се чуваат овие информации:
  - **String name**
    - Името на корисникот
  - **PrivateKey privateKey**
    - Приватниот клуч на корисникот
  - **PublicKey publicKey**
    - Јавниот клуч на корисникот
  - **MessageDigest digest**
    - Изгенерирање на хешот
  - **BigInteger alpha, BigInteger p**
    - Diffie-Hellman параметри

    - **Key sharedKey**
        - Заедничкиот изгенериран клуч кој се користи за понатамошно енкриптирање на комуникацијата
    - **BigInteger otherPersonDHValue**
        - Јавниот клуч на другата страна (другиот корисник со кој сакаме да комуницираме) кој се користи за изгенерирање на заедничкиот клуч (sharedKey)
- Функции:
    - **generateFirstMessage(BigInteger alpha, BigInteger p)**
        - Ја изгенерираме првата порака која се испраќа од Алис до Боб
    - **receiveFirstMessage(FirstMessage fm)**
        - функција која ја прима првата порака испратена од Алис до Боб
    - **sendCertificateRequest()**
        - се испраќа барање за сертификат од страна на корисникот до CA-то
    - **validateCertificate(Certificate certificate, PublicKey key)**
        - валидирање на сертификатот
    - **validateSecondMessage(SecondMessage sm, PublicKey key)**
        - валидирање на втората порака (која го содржи сертификатот на Боб)
    - **validateThirdMessage(ThirdMessage tm, PublicKey key)**
        - валидирање на третата порака (која го содржи сертификатот на Алис)

ОДГОВОРИ НА ПРАШАЊАТА:

3. Објаснете ги параметрите што ги има во барањата и одговорите. Зошто ни се потребни?

- Првото барање ги има параметрите: $\alpha,\ p,\ \alpha^x$
  - α – е примитивниот елемент, значи ни претставува основа на степенувањето што ќе се направи со приватниот клуч (y) на другата страна, значи α^y, што истовремено ни претставува и јавниот клуч на другата страна
  - p – ни претставува вредноста со која ќе се врши операцијата модул кога степенуваме
  - α^x – ни претставува јавниот клуч на првата засегната страна, и ќе се користи од другата страна за да се изгенерира заедничкиот клуч со операцијата $(\alpha^x)^y$
- Второто барање ги има параметрите: $\alpha^y, Cert_B, E_K(s_B\{\alpha^y, \alpha^x\})$
  - α^y – се испраќа за да може да се пресметува заедничкиот клуч, т.е. $(\alpha^y)^x$
  - $Cert_B$ – за да добиеме автентициран јавен клуч на Боб
  - $E_K(s_B\{\alpha^y, \alpha^x\})$ – за да се осигураме дека првиот параметар е така како што е испратен, а не беше сменет некаде на средина, и исто така да се осигураме дека прифатениот α^x од првата порака е така како што беше испратен, без да е сменет некаде на средина
- Третото барање ги има параметрите: $Cert_A, E_K(s_A\{\alpha^x, \alpha^y\})$
  - Првиот параметар е сертификатот на Алис, од каде може да се добие автентицираниот јавен клуч на Алис
  - Вториот параметар – за да се осигураме дека DH јавните клучеви изгенерирани од двете страни не беа сменети од некој на средина

4. Дали може да извршиме Man-in-the-Middle напад врз овој протокол? Зошто?

- НЕ МОЖЕ!
- Ако во првата порака е сменето нешто, тогаш кога ќе го примиме втората порака, со помош на вториот параметар ќе се дознае дека нешто не е во ред.
  - Јавниот клуч α^x примен од страна на Боб нема да се совпаѓа со α^x пресметуван од страна на Алис
- Ако во втората порака има некоја смена, тогаш ќе се детектира од енкриптираниот потпис на хешот на двете јавни клучеви, т.е. третиот параметар (кој ни нуди интегритет)
- Ако во третата порака има некоја смена, тогаш ќе се детектира од вториот параметар
  - Боб го има заедничкиот клуч така да ќе може да го декриптира вториот параметар, потоа ќе може да го декриптира потписот (ќе го добие јавниот клуч со помош на првиот параметар т.е. сертификатот), и ќе провери дали хешот на тие два јавни клучеви се совпаѓа со хешот на тие што веќе Боб ги има (од претходните пораки).
- Исто така и тоа што кај Сертификатот на Алис се содржат и DH параметрите, сигурни сме дека кај првата порака не беше сменето нешто, како на пр. јавниот клуч на Алис да се смени со некој слаб јавен клуч