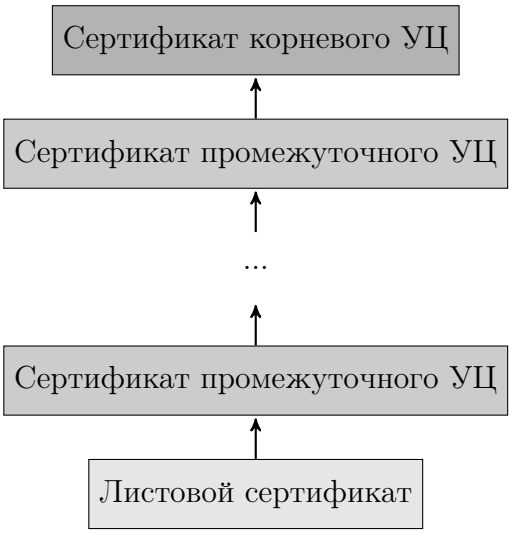
Лабораторная работа 7: «Создание и использование цифровых сертификатов» [до 1 июня]

1. О деталях реализации и средствах разработки

1.1. Цепочка сертификатов:



Проверка идентичности с помощью сертификата состоит в проверке: 1) является ли лицо, предъявившее сертификат для идентификации, его владельцем; 2) является ли действительным предъявляемый сертификат[1]. Каждый сертификат, кроме корневого, выпущен и подписан сертификатом предыдущего удостоверяющего центра[2].

1.2. Создание самоподписываемого сертификата[3] (используется в качестве корневого УЦ и считается надежным):

- Генерация пары ключей:

[1] Аналогия: чтобы идентифицировать себя, человек предъявляет паспорт, но паспорт должен быть действительным.

[2] Если промежуточный сертификат скомпрометирован, то его можно отозвать, не отзывая корневой сертификат и остальные промежуточные сертификаты, подписанные тем же корневым.

[3] Для локальной сети целесообразно создание собственного/корпоративного центра сертификации открытых ключей.

```
openssl genpkey -algorithm ED448 -out root_keypair.pem
```

- Создание простого запроса на подписание сертификата и просмотр запроса:

```
openssl req -new -subj "/CN=Root CA" -addext
↪  "basicConstraints=critical,CA:TRUE" -key root_keypair.pem
↪  -out root_csr.pem
```

```
openssl req -in root_csr.pem -noout -text
```

- Генерация сертификата (срок действия – 10 лет) и просмотр сертификата:

```
openssl x509 -req -in root_csr.pem -copy_extensions copyall
↪  -key root_keypair.pem -days 3650 -out root_cert.pem
```

```
openssl x509 -in root_cert.pem -noout -text
```

Сертификат самоподписываемый, поэтому поля `Issuer` и `Subject` одинаковы.

1.3. Создание несамоподписываемого сертификата (используется в качестве промежуточного УЦ и не считается надежным):

- Генерация пары ключей:

```
openssl genpkey -algorithm ED448 -out intermediate_keypair.pem
```

- Генерация запроса на подписание сертификата:

```
openssl req -new -subj "/CN=Intermediate CA" -addext
↪  "basicConstraints=critical,CA:TRUE" -key
↪  intermediate_keypair.pem -out intermediate_csr.pem
```

Поля `Issuer` и `Subject` отличаются.

- Выпуск сертификата промежуточного УЦ, подписанного закрытым ключом корневого сертификата, и его просмотр:

```
openssl x509 -req -in intermediate_csr.pem -copy_extensions
↪  copyall -CA root_cert.pem -CAkey root_keypair.pem -days
↪  3650 -out intermediate_cert.pem
```

```
openssl x509 -in intermediate_cert.pem -noout -text
```

- Выпуск листового сертификата (сообразно выпуску сертификата промежуточного УЦ), подписанного закрытым ключом сертификата промежуточного УЦ, и его вывод:

```
openssl genpkey -algorithm ED448 -out leaf_keypair.pem

openssl req -new -subj "/CN=Leaf" -addext
↪  "basicConstraints=critical,CA:FALSE" -key leaf_keypair.pem
↪  -out leaf_csr.pem

openssl x509 -req -in leaf_csr.pem -copy_extensions copyall
↪  -CA intermediate_cert.pem -CAkey intermediate_keypair.pem
↪  -days 3650 -out leaf_cert.pem
```

```
openssl x509 -in leaf_cert.pem -noout -text
```

Листовой сертификат не должен использоваться для выпуска других сертификатов, поэтому `CA:FALSE`.

1.4. Проверка сертификата (строится цепочка от проверяемого сертификата до надежного):

```
openssl verify -verbose -show_chain -trusted root_cert.pem
↪  -untrusted intermediate_cert.pem leaf_cert.pem
```

Флаг `-trusted` задает файл, содержащий один или несколько надежных сертификатов. Флаг `-untrusted` задает файл, содержащий один или несколько ненадежных сертификатов. Оба флага можно использовать несколько раз для задания нескольких файлов.

2. Постановка задачи

Напишите простое клиент-серверное приложение, в котором сервер выступает в качестве удостоверяющего центра, а клиенты могут обмениваться подписанными документами (с возможностью проверки подписей). Шифр: RSA.

3. Задания для подготовки к экзамену

1. xxx
2. xxx