Семинар 5. ЭЦП с эллиптическими кривыми

Изучить операции в группе точек эллиптической кривой для непрерывного и дискретного случаев:

https://andrea.corbellini.name/ecc/interactive/reals-add.html

---

Модуль ecdsa

В Python алгоритм ECDSA реализован в пакете **ecdsa**.

Пакет обеспечивает генерацию ключей, подпись, проверку и получение общего секрета для пяти наиболее распространённых эллиптических кривых с длиной ключа 192, 224, 256, 384 и 521 бит, в т.ч. кривой secp256k1, применяемой в Биткоине и других криптовалютах.

Установка пакета:

```
pip install ecdsa
```

В пакете определены классы **`SigningKey`** и **`VerifyingKey`** , представляющие, соответственно закрытый и открытый ключи алгоритма ECDSA.

Генерация ключей производится с помощью метода **`SigningKey.generate(curve)`**, параметром является используемая в алгоритме кривая.

Ключи могут быть сохранены в байтовую строку при помощи метода **`to_pem()`**, прочитаны из байтовой строки методом **`from_pem()`**.

Подписание сообщения закрытым ключом производится методом **`sign(msg)`** объекта **`SigningKey`**, где **`msg`** – подписываемое сообщение. Результатом работы метода является электронная подпись в виде байтовой строки.

Проверка подписи производится методом **`verify(sig, msg)`** объекта **`VerifyingKey`**, где **`sig`** – электронная подпись, **`msg`** – проверяемое сообщение. Если проверка успешна, возвращается **`True`**, если нет – генерируется исключение **`BadSignatureError`**.

Упражнения

1. Создать пару ключей secp256k1 и сохранить на диск.
2. Загрузить с диска закрытый ключ, подписать с его помощью сообщение и записать полученную подпись на диск.
3. Загрузить с диска публичный ключ, сообщение и подпись. Выполнить проверку подписи.

Генерация Биткоин-адреса

Биткоин-адрес генерируется на основе открытого ключа ECDSA с использованием следующего алгоритма:

1. В качестве закрытого ключа алгоритма ECDSA выбирается случайное 32-байтовое число от **`1`** до **`0xFFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFE BAAE DCE6 AF48 A03B BFD2 5E8C D036 4141`**.

2. Генерируется соответствующий открытый ключ с использованием реализации secp256k1 алгоритма ECDSA.

Открытый ключ записывается в несжатой форме, как два последовательно идущих 32-байтовых числа, которым предшествует байт со значением ***0x04*** *– признак несжатого открытого ключа.*

3. Вычисляется хеш SHA-256 от открытого ключа.

4. Вычисляется хеш RIPEMD-160 от хеш-значения предыдущего шага.

В настоящее время алгоритм RIPEMD-160 считается устаревшим. Его поддержка в библиотеке OpenSSL 3.0 прекращена и соответствующая функция в модуле ***hashlib*** *заблокирована. Для его использования в программах можно загрузить и установить отдельный модуль, например* ***ripemd-hash****:*

```
pip install ripemd-hash
```

5. К результату шага 4 дописывается слева номер версии адреса Биткоин (для основной сети Биткойн - значение **0x00**).

6. Результат шага 5 дважды хешируется с использованием функции SHA-256.

7. Первые 4 байта результата шага 6 дописываются справа к результату шага 5 – это контрольная сумма для проверки правильности адреса.

8. Полученное на шаге 7 значение переводится в кодировку base58 (*например, с использованием функции* **b58encode** *модуля* **base58**).

Данные для проверки:

Шаг	Величина	Значение
1.	Закрытый ключ ECDSA	**0x18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725**
2.	Несжатый открытый ключ ECDSA	**0x0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6**
3.	Хеш SHA-256 открытого ключа	**0x600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B408**
4.	Хеш RIPEMD-160 результата шага 3	**0x010966776006953D5567439E5E39F86A0D273BEE**
5.	Результат добавления номера версии адреса	**0x00010966776006953D5567439E5E39F86A0D273BEE**
6.	Двойное хеширование SHA-256 результата шага 5	**0xD61967F63C7DD183914A4AE452C9F6AD5D462CE3D277798075B107615C1A8A30**

7.	Результат шага 5 с контрольной суммой	**0x00010966776006953D5567439E5E39F86A0D273BEED61967F6**
8.	Результат шага 7 в кодировке Base-58	**16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM**