Стойкость MAC к коллизиям определяется исключительно в модели, при секретном ключе.  
При этом стойкость к коллизиям при известном ключе не гарантируется.

В данной лабораторной работе демонстрируется, что использование MAC с фиксированным ключом в качестве  
стойкой к коллизиям хэш функции - плохая идея.

В данной работе рассматривается truncated cbc mac, устроенный следующим образом - данные шифруются с использованием режима CBC,  
в качесте MAC используется половина последнего блока полученного шифртекста. Заметим, что в отличии от raw-CBC конструкции данная схема является стойким MAC. Используется PKCS7 дополнение до размера блока.

**NB**: хотя truncated cbc mac на AES теоретически стойкий, очевидно, что его размер составляет 64 бита, т.е. общая атака даёт лишь  
32 бита стойкости. Данный MAC используется в данной работе для простоты (в идеале нужно использовать другой стойкий MAC, с более сложной  
консрукцией или блочный шифр с большим размером блока). В рамках работы предположить, что 2^32 является  
"неперебираемым" для нас значением. Т.е. не нужно закладываться на то, что мы можем перебрать MAC. Атака должна работать и для  
блочных шифров с большим размером блока.

В качестве блочного шифра используется AES. Ключ в коде аутентичности фиксированный для задания.

В реальной жизни хэш функции часто используются для проверки целостности кода. В частности, в данном задании дана служба, которая  
проверяет целостность данных. Хэш функция должна гарантировать, что если данные были изменены, то изменится и хэш.  
Т.е. (например) имея доступ к возможности модификации исходного кода его нельзя изменить, не изменив хэш. Т.е. контроль хэша обеспечивает  
контроль целостности строк исходного кода. Покажем, что использовать MAC с фиксированным ключом в качестве хэша плохо, так как  
это позволяет модифицировать код, не изменяя при этом результирующий хэш (т.е. изменения невозможно отследить полученным хэшом  
=> это плохая хэш конструкция, как как получили коллизию).

## Задание

Дана REST служба с API указанным ниже. Дана строка кода на языке python

self.crypto.security\_parameter = configuration.settings.client.get\_security\_parameter()

Задача - получить строку кода, в которой переменной self.crypto.security\_parameter присваивается значение 32, и при этом  
имеющую такое же хэш значение (в качестве хэша используется truncated cbc mac с фиксированным ключом).

**ВАЖНО!** Все передаваемые сообщения должны быть предварительно закодированы с помощью BASE64.

Полученные сообщения так же должны быть декодированы из BASE64 в указанный тип.

**ВАЖНО!** Все строки кодируется с использование кодировки ASCII.

## Ход работы.

### Тестирование

<userId> = имя аккаунта на GitHub (или фамилия студента)

<challengeId> = 1

1. Проверить работоспособность контроллера с помощью метода GET <host>/api/CbcMacFixedKey
2. Получить текущий ключ с использованием метода GET <host>/api/CbcMacFixedKey/{userId}/{challengeId}/Key
3. Иодифицировать строку кода как указано в задании (см "Задача" выше), так, чтобы хэш новой строки остался равен хэшу изначальной строки.
4. Проверить, полученный результат с использованием метода POST <host>/api/CbcMacFixedKey/{userId}/{challengeId}/{mac}  
   следующим образом - для исходного и полученного сообщения получить результат "have valid mac" для одного и того же хэш значения.

### Сдача лр

шаги 1 - 4 этапа тестирования аналогично для 20 различных <challengeId>.

## Описание API

Rest запросы, в заголовке выстален Content-Type: application/json; charset=utf-8.

### Описание методов

## GET <host>/api/CbcMacFixedKey

Проверка работоспособности контроллера. Возвращает operating. Ответ не кодируется в BASE64.

|  |  |
| --- | --- |
| Параметр | Описание |
| <host> | имя хоста веб службы |

## POST <host>/api/CbcMacFixedKey/{userId}/{challengeId}/{mac}

Вычисляет хэш от полученных данных.  
Возвращает строку, содержащую информацию о том, совпадает ли хэш данных с предоставленным хэш значением.

|  |  |
| --- | --- |
| Параметр | Описание |
| <host> | имя хоста веб службы |
| <userId> | идентификатор студента |
| <challengeId> | идентификатор задания |
| <mac> | проверяемое хэш значение |

## GET <host>/api/CbcMacFixedKey/{userId}/{challengeId}/Key

Получает ключ от текущего задания, который используется для вычисления хэш функции.

|  |  |
| --- | --- |
| Параметр | Описание |
| <host> | имя хоста веб службы |