**Аутентификация по программе SKEY (на основе однонаправленных функций)**

**Теоретические сведения.**

S/Key представляет собой систему генерирования одноразовых паролей на основе стандартов MD5. Она предназначена для борьбы с так называемыми “повторными атаками”, когда хакер подслушивает канал, выделяет из трафика аутентификатор пользователя и его пароль и в дальнейшем использует их для несанкционированного доступа.

Принцип работы:

1) Алиса задает свое случайное число.

2) значения . сохраняет в незашифрованном виде значение в регистрационной базе данных напротив имени Алиса

3)

Эти обмены проходят по защищенному каналу. В дальнейшем при каждом входе в систему проводится следующий протокол аутентификации.

Протокол

1) , при первом входе в систему Алиса вводит свое имя и значение .

2) . вычисляет и сравнивает его со значением , которое хранится в регистрационной базе. Если значения совпадают, подлинность Алисы подтверждается. Затем H заменяет в базе данных на .

3) удаляет из своего списка .

Далее при каждом входе в систему Алиса вводит последнее не вычеркнутое число из своего списка, например . вычисляет и сравнивает его со значением , хранящемся в базе данных. Так как каждый номер используется только один раз, Ева не сумеет получить никакой полезной информации. Точно так же база данных бесполезна и взломщику. Разумеется, когда Алиса исчерпает числа из своего списка, ей придется заново инициализировать систему

**Алгоритм**

1) Происходит регистрация пользователя. Пользователь вводит свое имя, количество одноразовых паролей и длина начального ключа. Генерируется секретный ключ . На основе ключа происходит генерация n одноразовых паролей для входа.

2)Генерация паролей будет происходить на основе однонаправленного алгоритма хеширования MD5. Первым паролем будет , вторым - и т.д. Алгоритм хеширования будет применяться

3)Если пользователь уже авторизован в системе, он вводит свое имя и пароль. Введенный пароль хешируется и сравнивается с паролем (на сервере). После успешной авторизации у нас обновляется пароль на хосте.

**Листинг кода.**

import os  
import sys  
import hashlib  
  
def skey\_registr():  
 print("Этап регистрации")  
 print("Введите имя пользователя: ", end = '')  
 name = input()  
 if os.path.exists(name + '.txt'):  
 print("Пользователь с таким именем уже существует")  
 sys.exit(-1)  
 dict\_hash = []  
 print("Введите секретный ключ: ", end = '')  
 key = input()  
 for i in range(50):  
 key = hashlib.md5(key.encode("utf8")).hexdigest()  
 dict\_hash.append(key)  
 f = open(name + '.txt', 'w')  
 for i in dict\_hash:  
 f.writelines(i + '\n')  
  
  
def skey\_login():  
 print("Этап аутентификации")  
 print("Введите ваше имя: ", end = '')  
 name = input()  
 if not os.path.exists(name + '.txt'):  
 print("Вас нет в регистрационной базе")  
 sys.exit(-1)  
 print("Введите пароль: ", end='')  
 password = input()  
 if os.stat(name +".txt").st\_size == 0:  
 print("Необходимо обновить регистрационную базу")  
 sys.exit(-1)  
 f = open(name + '.txt', 'r')  
 lines = []  
 for i in f.readlines():  
 i = i[:-1]  
 lines.append(i)  
 f.close()  
 if password == lines[0]:  
 print("Вы успешно зашли!")  
 f = open(name + '.txt', 'w')  
 lines = lines[1:]  
 for i in lines:  
 f.writelines(i + '\n')  
 f.close()  
 else:  
 print("Пароль неверный!")  
 sys.exit(-1)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 action = input("Введите 0 - для регистрации пользователя, 1 - для аутентификации пользователя, 2 - для выхода: ")  
 if action == "0":  
 skey\_registr()  
 elif action == "1":  
 skey\_login()  
 elif action == "2":  
 sys.exit(-1)