**Задание к выполнению лабораторной работы  
«АУТЕНТИФИКАЦИЯ И ШИФРОВАНИЕ В GSM»**

**Подготовка к выполнению работы:**

1. Скачать и установить демо-версию программы RuimTool «2G + 3G/4G Authentication». Скачать можно с официального сайта <http://www.ruimtools.com/products.php>
2. Найти в Интернете и скачать таблицу ASCII.

**Задание:**

*Произвести расчет SRES и Kc с помощью программы 2G + 3G/4G Authentication, после чего закодировать текстовое сообщение ключом Kc путем простого сложения по модулю 2.*

1) Ключ KI (16 байт) зависит от варианта:

KI = 8749 8496 49DF A8ХХ F278 1C87 D87B B5XX

где XX – две последние цифры пароля

2) Последовательность RAND нужно сгенерировать самостоятельно. Требования следующие:

a.   Длина 16 байт.

b. Записать в виде последовательности из 32 шестнадцатеричных символов.

c.   Шестнадцатеричные символы: 0-9, A,B,C,D,E,F.

d. Ключ KI может послужить примером, но RAND должен отличатся от KI.

3) С помощью программы «2G + 3G/4G Authentication» вычислить SRES и Kc. Пример показан на рисунке 1.

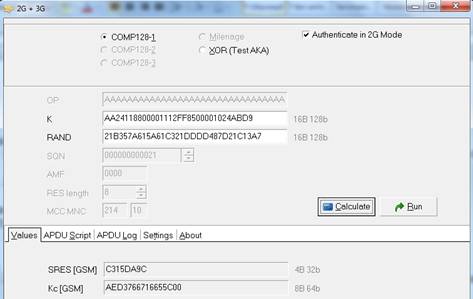


Рисунок 1 – Пример вычисления SRES и Kc.

4) Пояснить, зачем нужны два этих значения.

5) Напишите произвольный текст на английском 10-15 символов, вначале текста вставьте номер своего варианта, например «22 hello world». Можно использовать пробелы.

6) С помощью таблицы ASCII запишите представление данного текста в шестнадцатеричном и двоичном видах. Пробел тоже считается за знак, в таблицах он представлен как спецсимвол «Space». Например:

Текст «22 hello world»

В шестнадцатеричном виде: 32322068656c6c6f20776f726c64

В двоичном виде: 0011001000110010001000000110100001100101011011000110110001101111001000000111011101101111011100100110110001100100

7) Запишите ключ Kc в двоичном виде, например:

Kc = AED3766716655C00.

В двоичном виде: 1010111011010011011101100110011100010110011001010101110000000000

8) Теперь зашифруйте текст ключом Kc. Так как у нас под рукой нет алгоритма шифрования A5, то зашифруем текст простым побитным сложением ключа и текста. Для этого выполните побитную операцию XOR (сложение по модулю 2) в двоичном виде между Kc и вашим текстом. Так как Kc будет короче, чем текст, то используйте Kc циклически. То есть если длины Kc не хватило, просто начинайте использовать его сначала. Пример:

Текст «22 hello world»:

0011001000110010001000000110100001100101011011000110110001101111001000000111011101101111011100100110110001100100

Ключ Kc:

1010111011010011011101100110011100010110011001010101110000000000101011101101001101110110011001110001011001100101

Жёлтым выделен исходный ключ, а зеленым выделено его повторение, для того чтобы текст и ключ Kc были одинаковые по длине.

Результат XOR (зашифрованный текст):

1001110011100001010101100000111101110011000010010011000001101111100011101010010000011001000101010111101000000001

9) Переведите результат из двоичного вида в шестнадцатеричный. Продолжаем наш пример:

Зашифрованный текст в шестнадцатеричном виде:

9c e1 56 0f 73 09 30 6f 8e a4 19 15 7a 01

10)  Сравните исходный текст и зашифрованный в шестнадцатеричном виде. Сделайте выводы. Можно ли по таблице ASCII перевести зашифрованный текст из шестнадцатеричного вида в символы?