

Лабораторна робота № 2 Симетричне шифрування. Алгоритм AES

Група КН-М922в

Автор Савичев О.В.

Мета

Дослідити принципи роботи симетричного шифрування на прикладі алгоритму AES.

Завдання

- Реалізувати алгоритм симетричного шифрування AES (будь-якої версії - 128 або 256).
- Довести коректність роботи реалізованого алгоритму шляхом порівняння результатів з існуючими реалізаціями (напр. сайтом-утилітою <https://cryptii.com>).

Хід роботи

Advanced Encryption Standard (AES), також відомий під назвою Rijndael — симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт), фіналіст конкурсу AES і прийнятий як американський стандарт шифрування урядом США. Вибір припав на AES з розрахуванням на широке використання та активний аналіз алгоритму, як це було із його попередником, DES. Державний інститут стандартів і технологій (англ. National Institute of Standards and Technology, NIST) США опублікував попередню специфікацію AES 26 жовтня 2001 року, після п'ятилітньої підготовки. 26 травня 2002 року AES оголошено стандартом шифрування. Станом на 2009 рік AES є одним із найпоширеніших алгоритмів симетричного шифрування.

Головний метод шифрування. Розбиває повідомлення на блоки, та шифрує кожен блок окремо

```
public byte[] Encrypt(byte[] buf)
{
    using (var bout = new MemoryStream())
    {
        using (var bin = new MemoryStream(_PadBuffer(buf, BlockSize)))
        {
            var block = new byte[BlockSize];
            int c;
            byte[,] state;
            while ((c = bin.Read(block, 0, BlockSize)) > 0)
            {
                state = LoadState(block);
                _EncryptBlock(state, _ExpandedKey);
                bout.Write(DumpState(state), 0, c);
            }
        }
        return bout.ToArray();
    }
}
```

Шифрування блоку

```
void _EncryptBlock(byte[] state, uint[] key)
{
    _AddRoundKey(state, _GetUIntBlock(key));
    for (int i = 1; i <= Rounds; i++)
    {
        _SubBytes(state);
        _ShiftRows(state);
        if (i != Rounds)
            _MixColumns(state);
        _AddRoundKey(state, _GetUIntBlock(key, i));
    }
}
```

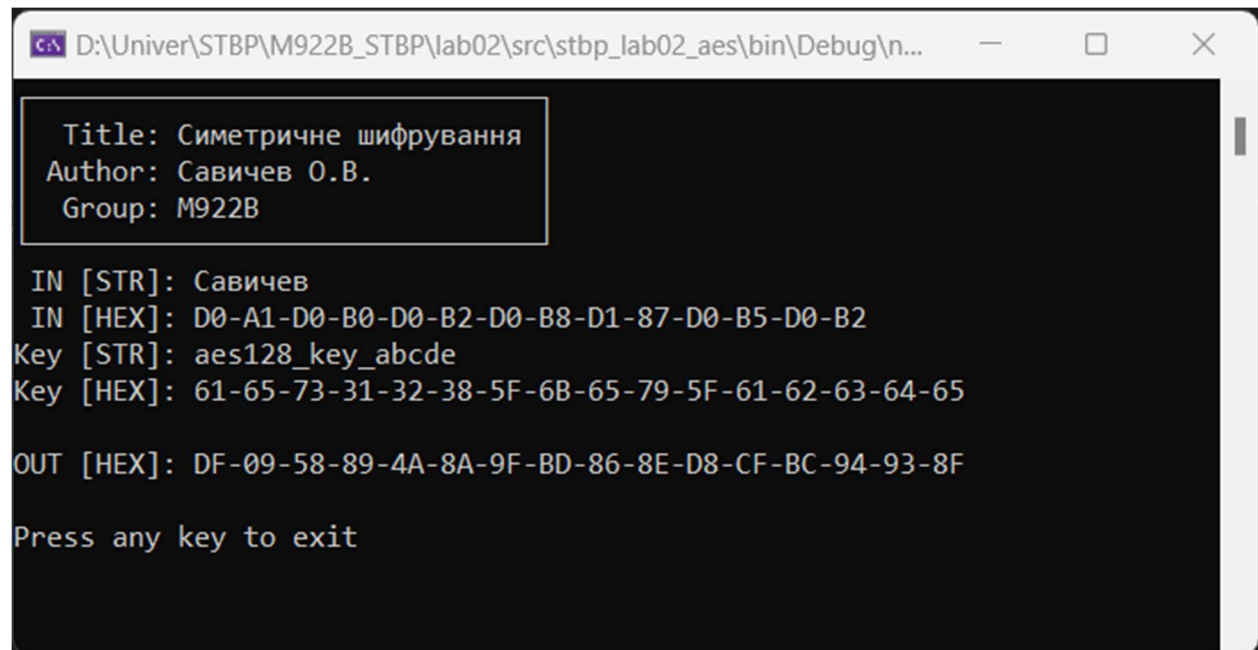
Обчислення раундових ключів для всіх раундів

```
uint[] ExpandKey(byte[] key)
{
    int init = 4;
    uint[] expandedKey = new uint[ExpandedKeyWords];
    for (int i = 0; i < init; i++)
        expandedKey[i] = _GetWord(key, i * WordSize);

    int iteration = 1;
    int counter = 0;
    for (int i = init; i < ExpandedKeyWords; i += WordSize)
    {
        var tmp = expandedKey[i - 1];
        tmp = _KeySchedule(tmp, iteration);
        iteration++;
        counter++;
        for (int j = 0; j < WordSize; j++)
        {
            tmp ^= expandedKey[i - init + j];
            expandedKey[i + j] = tmp;
        }
    }
    return expandedKey;
}
```

Результати роботи

Результати роботи моєї реалізації



```
D:\Univer\STBP\M922B_STBP\lab02\src\stbp_lab02_aes\bin\Debug\n...
Title: Симетричне шифрування
Author: Савичев О.В.
Group: M922B

IN [STR]: Савичев
IN [HEX]: D0-A1-D0-B0-D0-B2-D0-B8-D1-87-D0-B5-D0-B2
Key [STR]: aes128_key_abcde
Key [HEX]: 61-65-73-31-32-38-5F-6B-65-79-5F-61-62-63-64-65

OUT [HEX]: DF-09-58-89-4A-8A-9F-BD-86-8E-D8-CF-BC-94-93-8F

Press any key to exit
```

Порівняння з онлайн-сервісом обчислення AES-128

Enter Plain Text to Encrypt -	<input type="text" value="Савичев"/>
Select Mode	<input type="button" value="ECB"/>
Key Size in Bits	<input type="button" value="128"/>
Enter Initialization Vector -	<input type="text"/>
Enter Secret Key -	<input type="text" value="aes128_key_abcde"/>
Output Text Format	<input type="button" value="Hex"/>

Висновок

Дослідив принципи роботи симетричного шифрування на прикладі алгоритму AES-128.