Контрольна робота

З предмету Основи Криптології

Волошина Владислава Руслановича

Групи ІПС-31

Варіант 11

1. Шифр AES:

AES або Advanced Encryption Standard – це симетричний алгоритм блочного шифрування, що шифрує блоки по 128 біт за допомогою ключів довжиною у 128, 192 та 256 бітів. Для пояснення принципу роботи алгоритму нехай будемо використовувати 128-бітний ключ.

Спочатку ключ розгортається, тобто для 128-бітного ключа $\boldsymbol{K}$ будується $\boldsymbol{R+1}$ 128-бітних ключів, де R позначає кількість раундів (R=10), при чому кожен з ключів складається з 32-бітних слів W_i , де $0 \leq i \leq 4R-1$. Слова отримуються за формулою:

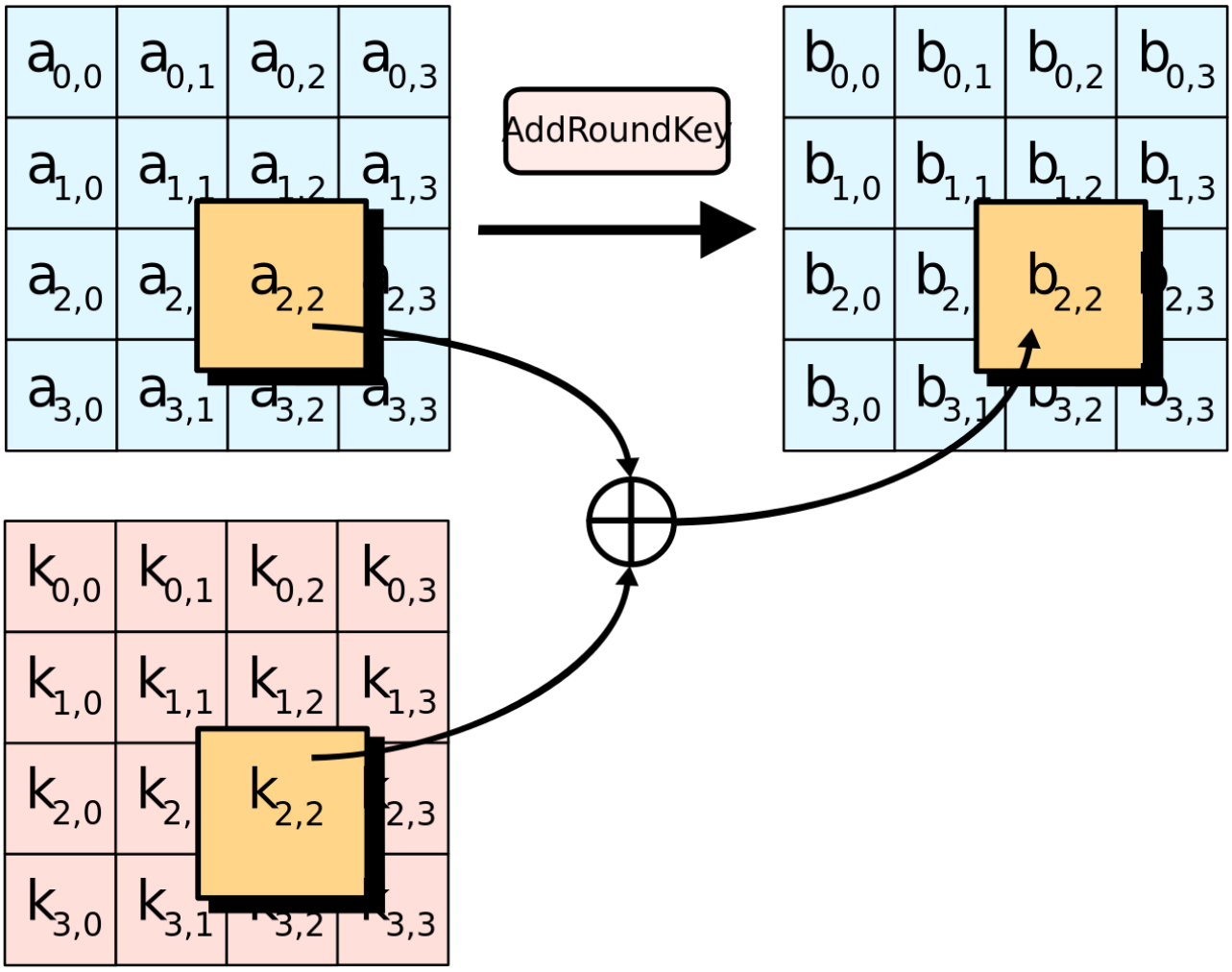
$W_i = K_i$, при $i < N$;
$W_i = W_{i-N} \oplus \text{SubWord}(\text{RotateWord}(W_{i-1})) \oplus \text{rcon}_i/N$, при $i \geq N$ і $i \equiv 0 (\text{mod} N)$;
$W_i = W_{i-N} \oplus \text{SubWord}(W_{i-1})$, при $i \geq N$, $N \geq 6$ і $i \equiv 4 (\text{mod} N)$;
$W_i = W_{i-N} \oplus W_{i-1}$, інакше.
Де N – це кількість 32-бітних слів у ключі, RotateWord – це функція, що виконує циклічний зсув слова вліво на один біт, SubWord – S перетворення для кожного з байтів,
rcon_i - 32-бітна константа для і-го раунду $\text{rcon}_i=[\text{rc}_i \quad 00_{16} \ 00_{16} \ 00_{16}]$.

Шифрування 128-бітного блока полягає у наступному:

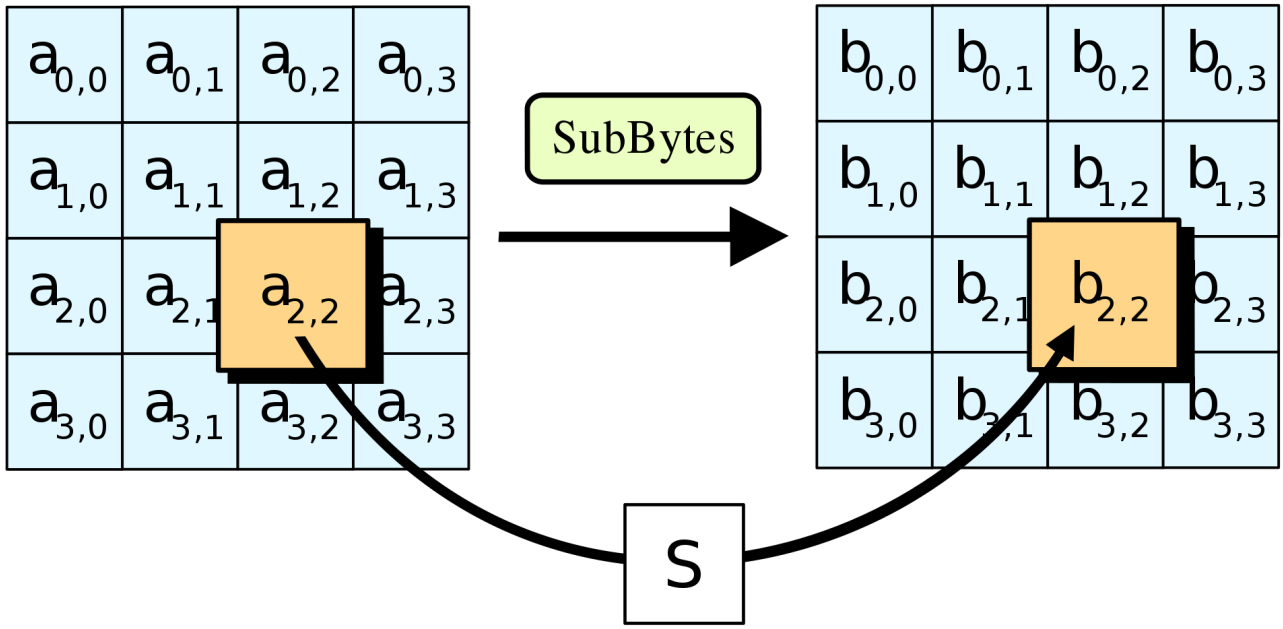
1. По стовпчиках формується матриця 4x4 з байтів блоку.
2. Береться ключ 0-го раунду за допомогою функції AddRoundKey.
3. R−1 раундів операцій:
    1. SubBytes
    2. ShiftRows
    3. MixColumns
    4. AddRoundKey
4. Останній раунд, який складається з операцій:

1. SubBytes
2. ShiftRows
3. AddRoundKey

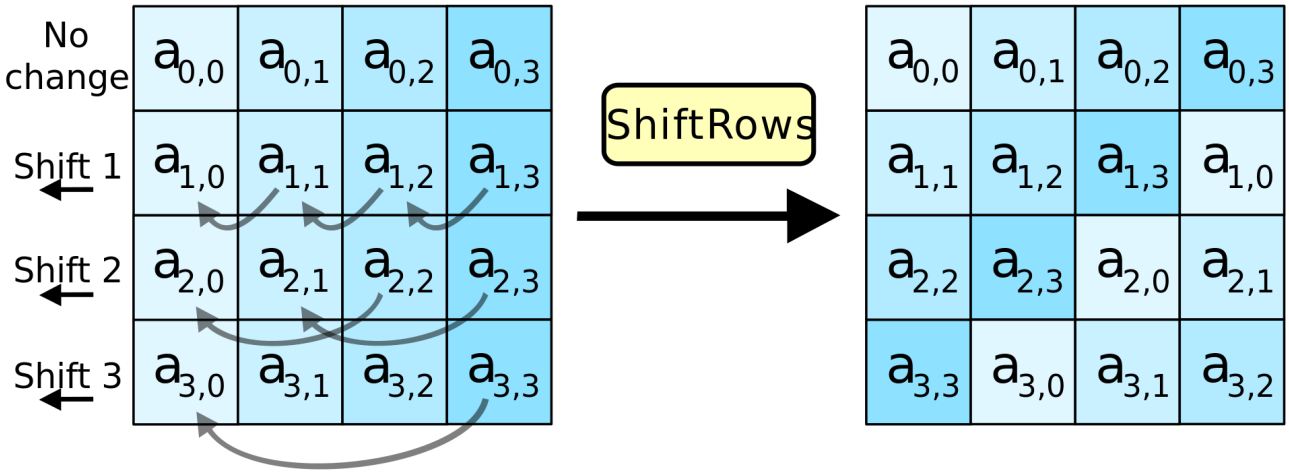
Операція AddRoundKey полягає у XOR-і стану-матриці та ключа для цього раунду.



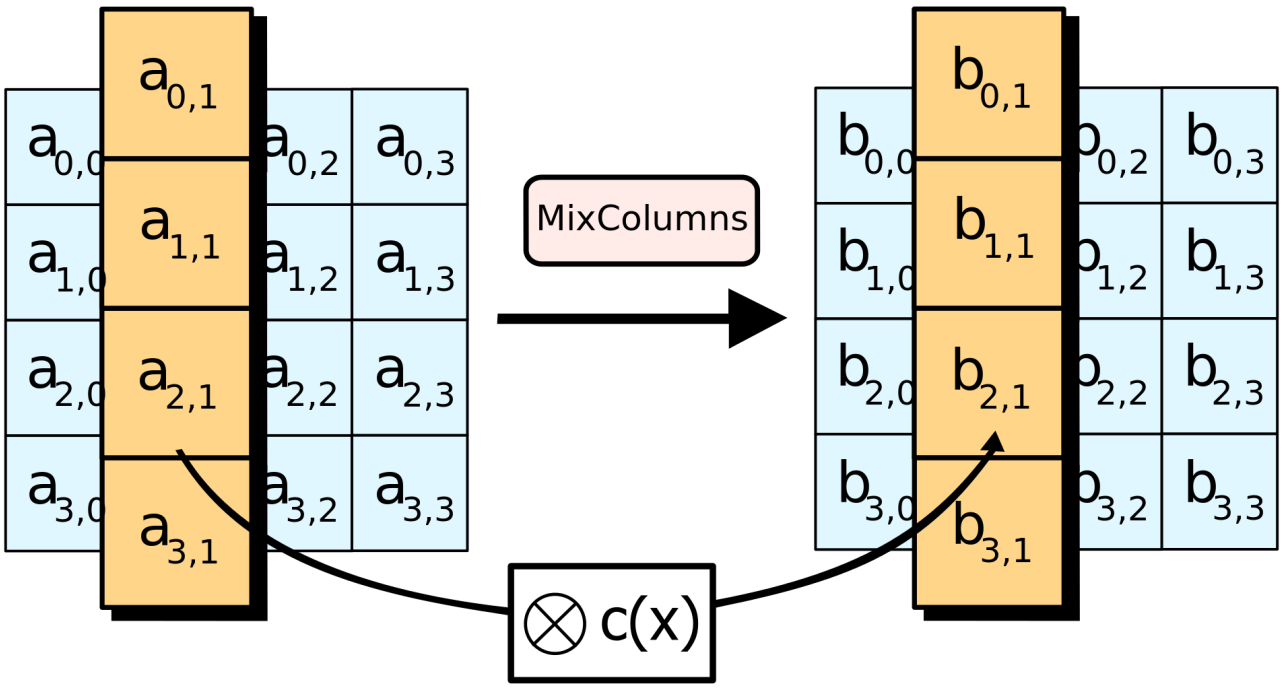
Операція SubBytes полягає у застосуванні S перетворення до кожного елементу матриці, це перетворення можна задати константним масивом.



Операція ShiftRows циклічно сдвигає байти у рядках матриці вліво, на 0 для першого рядка, на 1 для другого, на 2 для третього та на 3 для четвертого рядка.



Операція MixColumns полягає у множенні кожного стовпчика матриці на певну константну матрицю зліва, причому операцію додавання замінює XOR, а для операції множення біти трактуються як коефіцієнти полінома 7-го степеня, ці поліноми множаться по модулю незвідного полінома $x^8+x^4+x^3+x+1$.



Блоки оброблюються довільним з режимів шифрування.

Дешифрування тексту полягає у дешифруванні 128-бітних блоків відповідно до обраного режиму шифрування. Після чого видаляються біти вирівнювання.

Алгоритм дешифрування 128-бітного блоку полягає у використанні інверсних операцій у оберненому порядку:

1. По стовпчиках формується матриця 4x4 з байтів блоку.
2. AddRoundKey (береться ключ останнього раунду)

   InvShiftRows

   InvSubBytes
3. R−1 раундів операцій (ключі у зворотному порядку):
    1. AddRoundKey
    2. InvMixColumns
    3. InvShiftRows
    4. InvSubBytes
4. AddRoundKey (використовується ключ 0-го раунду).

   InvSubBytes є застосуванням оберненого до S перетворення, і теж задається константим масивом.

InvShiftRows є тою самою, що і ShiftRows, тільки зсув відбувається вправо.

InvMixColumns є тою самою, що MixColumns, тільки матриця інша.

2. Протокол SSL:

Криптографічний протокол SSL (Secure Sockets Layer) служить для встановлення безпечного з'єднання між клієнтом і сервером. Він дозволяє автентифікацію та безпечну передачу даних в мережі з використанням криптографічих засобів.

SSL працює поверх TCP\IP та нижче протоколів вищого рівня, таких як HTTP або IMAP. Протокол дозволяє двом машинам встановити зашифроване з'єднання.

Функції SSL-автентифікації:

1. SSL-автентифікація дозволяє користувачеві підтвердити автентичність сервера. Програмне забезпечення SSL-клієнта дає можливість використовувати стандартні методи криптографії з відкритим ключем для перевірки сертифікатів сервера та відкритих ID.
2. SSL-автентифікація клієнта дозволяє серверу підтверджувати справжність користувача. Використовуючи ті самі методи, що й при автентифікації сервера, SSL може перевірити, що сертифікат клієнта є дійсним і виданим ЦС.
3. Вся інформація, що передається між клієнтом та сервером через SSL-з'єднання шифрується. Крім того, всі дані, що передаються через шифроване з'єднання SSL, захищені механізмом виявлення. Протокол SSL складається з двох підпротоколів: протокол SSL-запису та рукостискання. Протокол запису визначає формат, який використовується при передачі даних. Протокол SSL запускає рукостискання з використанням протоколу SSL-запису для обміну серіями повідомлень між сервером та клієнтом під час встановлення першого з'єднання. Цей обмін повідомленнями призначений для забезпечення наступної послідовності дій:
    - автентифікація сервера до клієнта,

- дозвіл клієнту та серверу вибрати криптографічні алгоритми або шифри, які вони підтримують,
- додаткова автентифікація клієнта на сервері,
- використання протоколу розподілу для створення загального секрету,
- створення шифрованого SSL-з'єднання.

Процедура рукостискання:

Під час рукостискання клієнт та сервер домовляються про різні параметри, які будуть використані для забезпечення безпеки з'єднання:

1. Рукостискання розпочинається, коли клієнт підключається до SSL-сервера. Запит безпечного з'єднання містить список підтримуваних шифрів та хеш-функцій.
2. З цього списку сервер обирає найсильніший шифр та хеш-функцію, яку він також підтримує, та повідомляє клієнтів про свій вибір.
3. Сервер надсилає цю відповідь у вигляді цифрового сертифікату (X.509). Сертифікат зазвичай містить інформацію про ім'я сервера, довірений центр сертифікації і відкритий ключ шифрування сервера. Перш ніж продовжити, клієнт може зв'язатися з сервером, який видав сертифікат (довіреного центру сертифікації) та переконатися в його справжності.
4. Для створення ключів сеансу використовується безпечне з'єднання. Клієнт шифрує випадкове число за допомогою відкритого ключа сервера та надсилає туди результат. Сервер може розшифрувати його за допомогою свого закритого ключа.
5. З випадкового числа обидві сторони створюють ключові дані для шифрування та розшифрування.

Далі сервер та клієнт обмінюються шифрограмами.

Розглянемо, що відбувається при рукостисканні докладніше.

1. Клієнт надсилає повідомлення ClientHello на відповідний порт, вказуючи найостаннішу останню версію SSL-протоколу, що підтримується, випадкове

число і список підтримуваних методів шифрування та стиснення.

2. Сервер відповідає повідомленням ServerHello, що містить обрану сервером версію протоколу, випадкове число, надіслане клієнтом, відповідні алгоритми шифрування та стиснення з наданого клієнтом списку.

3. Сервер надсилає повідомлення Certificate, яке містить цифровий сертифікат сервера (залежно від алгоритму шифрування, цей етап може бути пропущений).

4. Сервер може запитати сертифікат у клієнта, в такому разі з'єднання буде взаємоавтентифіковано.

5. Сервер надсилає повідомлення ServerHelloDone, що ідентифікує закінчення рукостискання.

6. Клієнт відповідає повідомленням ClientKeyExchange, яке містить відкритий ключ PreMasterSecret або нічого (залежить від алгоритму шифрування).

7. Клієнт та сервер, використовуючи ключ PreMasterSecret та випадково згенеровані числа, обчислюють загальний секретний ключ. Решта інформації про

ключ буде отримана із загального секретного ключа (і згенерованих клієнтом та сервером випадкових значень).

Останні два пункти фактично є протоколом Діффі-Геллмана.

8. Клієнт надсилає повідомлення ChangeCipherSpec, яке вказує на те, що вся наступна інформація буде зашифрована встановленим в процесі рукостискання алгоритмом, використовуючи загальний секретний ключ.

9. Клієнт надсилає повідомлення Finished, яке містить хеш-код і MAC-код автентифікації повідомлення, згенеровані на основі попередніх повідомлень рукостискань.

10. Сервер намагається розшифрувати Finished-повідомлення клієнта та перевірити хеш-код та MAC. Якщо процес розшифровки або перевірки не вдається, рукостискання вважається невдалим і з'єднання має бути обірвано.

11. Сервер надсилає ChangeCipherSpec та зашифроване Finished-повідомлення, в свою чергу клієнт теж виконує розшифровку та перевірку.