Тема:
Разработка модуля криптографической защиты.

Цель работы – разработка блока криптографической системы, который способен осуществлять операции шифрования информации.

Проблема защиты информации от несанкционированного доступа заметно обострилась в связи с широким распространением компьютерных сетей (особенно глобальных). Защита информации необходима для уменьшения вероятности разглашения, утечки, умышленного искажения, утраты или уничтожения информации, представляющую определенную ценность.

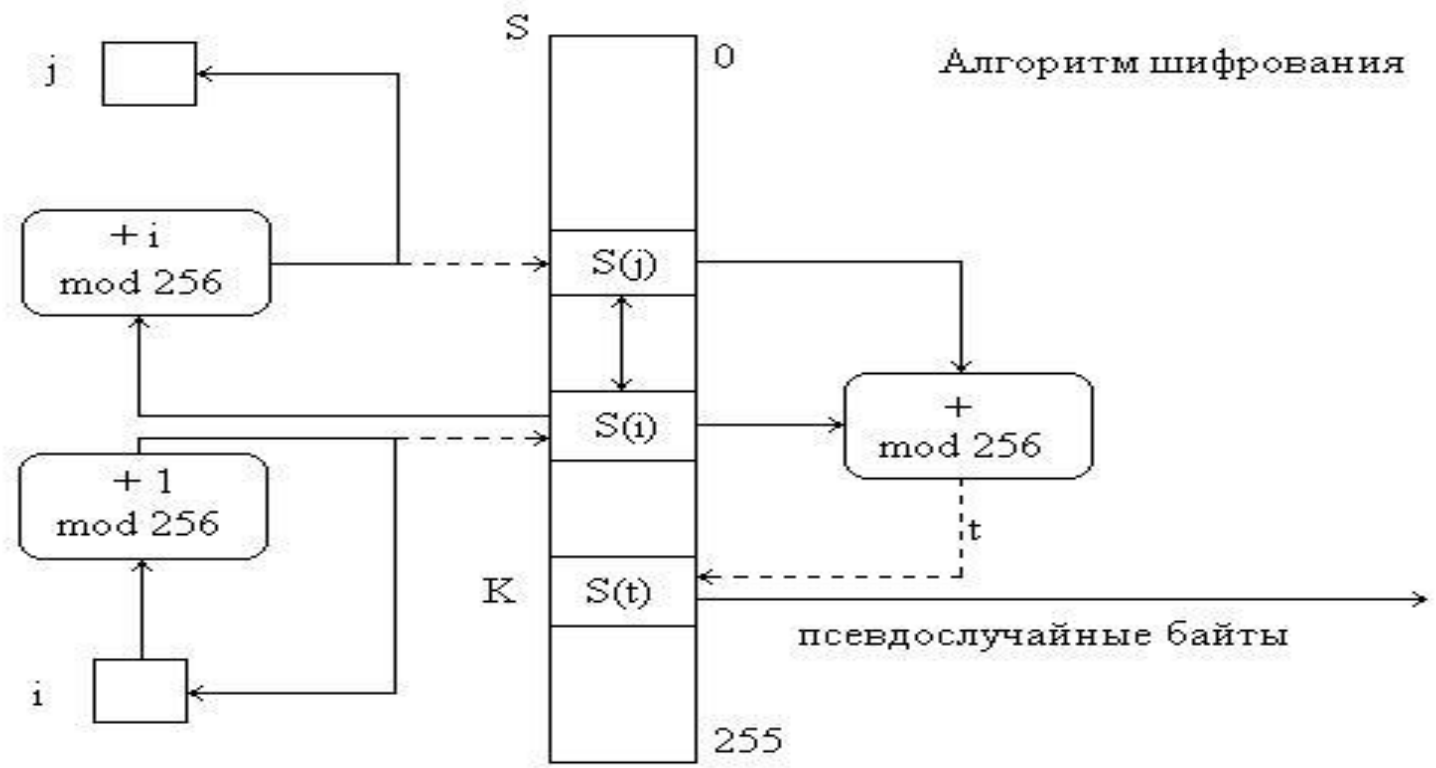
Практическая значимость работы заключается в возможности применения разработанной программы пользователями и организациями для защиты информации.

Выбор реализации:

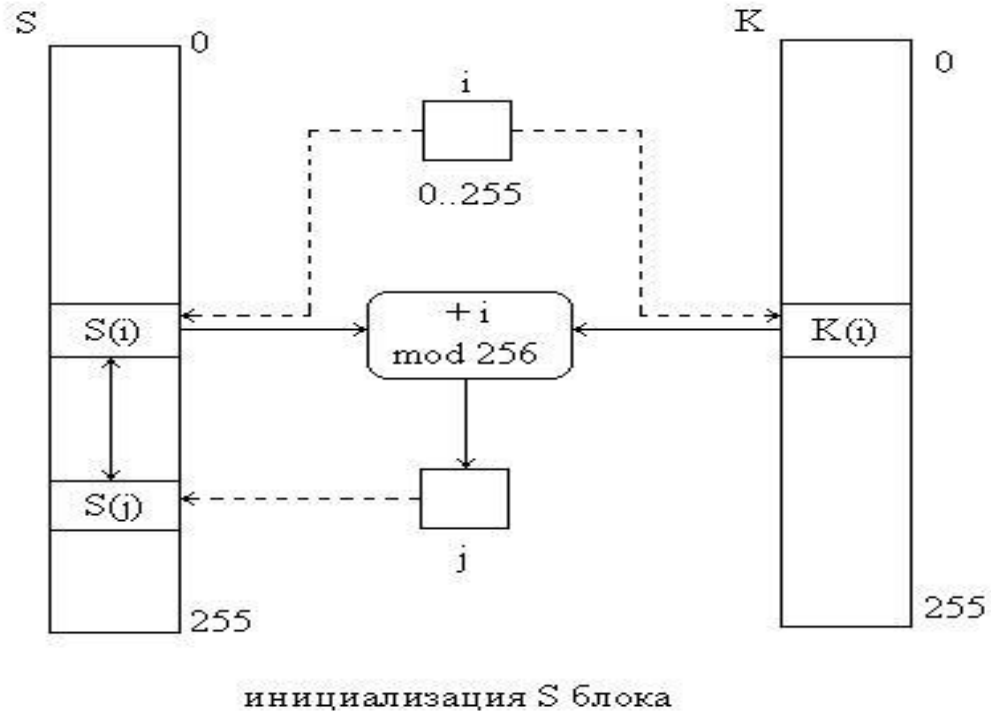
- *программная реализация модуля криптографической защиты* (программные средства шифрования легко копируются, они просты в использовании, их нетрудно модифицировать в соответствии с конкретными потребностями);
- *симметричное поточное шифрование* (предполагается, что известен алгоритм шифрования, зашифрованный текст, но неизвестен ключ шифрования; шифрование проводится над каждым битом (байтом) открытого текста)
- *алгоритм RC4* (потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и и TLC, алгоритмах обеспечения безопасности беспроводных сетей с WEP и с WPA).

Алгоритм шифрования **RC4**

- Строится на основе генератора псевдослучайных битов
- Длина ключа от 40 до 2048 бит
- Генерируемые биты имеют равномерное распределение
- Поток ключей не зависит от открытого текста



i = (i+1) mod 256;
j = (i+ S(i))) mod 256;
меняем местами S(i) и S(j);
t = (S(i) + S(j)) mod 256;
K = S(t).
Байт K используется в операции XOR с открытым текстом для получения шифротекста или в операции XOR с шифротекстом для получения открытого текста.



инициализация S блока

В начале происходит инициализация блока S . Сначала нужно заполнить его линейно: S(0)=0;....;S(255)=255. Затем заполнить ключом другой 256-байтовый массив, при небходимости для заполнения всего массива повторяя ключ: K(0),...,K(255),j=0.

for i=0 to 255:
j=(j+S(i)+K(i)) mod 256 ;
меняем местами S(i) и S(j).

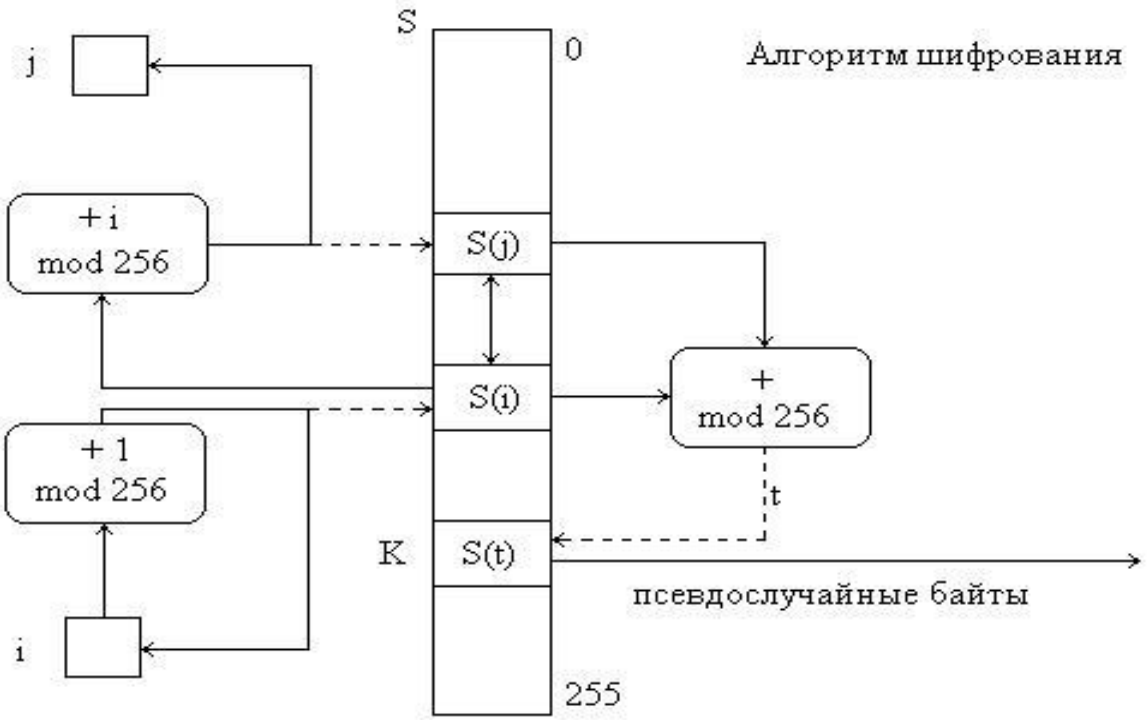
Алгоритм устойчив к дифференциальному и линейному криптоанализу, он в высокой степени нелинеен.

S -блок медленно изменяется при использовании:

i обеспечивает изменение каждого элемента, а j - что элементы изменяются случайным образом.

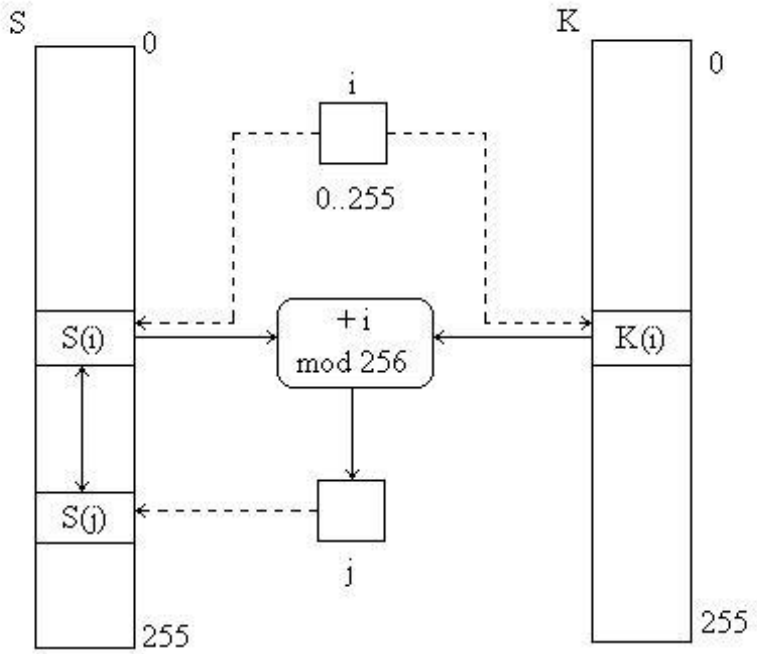
Пример:

Для примера возьмем ключ «абв», соответствующие ASCII-коды – 160,161,162.



j=0.
i = 0;
j = (0 + S(0) + K(0)) mod 256 = 160;
Меняем местами S(0) и S(160);
i = 1;
j = (1 + S(1) + K(1)) mod 256 = 163;
Меняем местами S(1) и S(163);
и т.д.
Получаем последовательность S(0) = 160, S(1) = 163, S(2) = 166 и т.д.

Затем используем полученную последовательность для инициализации блока S, который используется непосредственно для создания ключа шифрования. Далее установим счетчики i и j в 0. Затем, следуя по алгоритму, получаем:



инициализация S блока

i = (0+1) mod 256 = 1;
j = (1+S(1)) mod 256 = 164;
Меняем местами S(1) и S(164);
t = (S(1) + S(164)) mod 256 = (163 + 164) mod 256 = 71;
K = S(71).
i = (1+1) mod 256 = 2;
j = (2+S(2)) mod 256 = 168;
Меняем местами S(2) и S(168);
t = (S(2) + S(168)) mod 256 = (166 + 168) mod 256 = 78;
K = S(78).
и т.д.
Получаем последовательность байтов ключа K. Она разбивается на биты. Далее производим операцию XOR с битами текста.

Основные достоинства и недостатки RC4 :

Основные преимущества шифра:

- Простота и логичность каждого шага;
- высокая скорость работы алгоритма;
- переменный размер ключа (40-2048 бит).

RC4 довольно уязвим, если:

- используются не случайные или связанные ключи;
- один ключевой поток используется дважды.

Презентация окончена.

Спасибо за внимание!