Шифр Шамира: Алгоритм кодирования:

1. Задается открытый ключ (n, e), где n - произведение двух разных простых чисел, а e - натуральное число, взаимно простое с (n).
2. Представляется сообщение в виде числа m, которое должно быть меньше n.
3. Вычисляется шифрованный текст c по формуле c = m^e mod n, где "mod" обозначает операцию взятия остатка от деления.

Алгоритм декодирования:

1. Задается закрытый ключ (n, d), где d - число, обратное по модулю (n) для e.
2. Полученный шифрованный текст c декодируется по формуле m = c^d mod n.

Плюсы:

* Шифрование и дешифрование осуществляются с использованием открытого ключа, что обеспечивает безопасность передачи данных.
* Алгоритм является относительно простым в реализации.

Минусы:

* Алгоритм требует больших вычислительных мощностей при использовании больших ключей.
* Не обладает абсолютной защитой от всех видов атак.

Шифр Эль Гамаля: Алгоритм кодирования:

1. Задается открытый ключ (p, g, y), где p - большое простое число, g - примитивный элемент поля вычетов по модулю p, а y - степень элемента g по модулю p.
2. Представляется сообщение в виде числа m, которое должно быть меньше p.
3. Задается случайное число k, взаимно простое с p-1.
4. Вычисляются две части шифрованного текста:
   1. a = g^k mod p
   2. b = y^k \* m mod p

Алгоритм декодирования:

1. Полученный шифрованный текст представлен в виде a и b.
2. Вычисляется расшифрованное сообщение по формуле m = b \* a^(p-1-x) mod p, где x - случайное число, называемое закрытым ключом.

Плюсы:

* Использует примитивные элементы поля, что обеспечивает хорошую степень защиты.
* Сложность расшифрования зависит от сложности задачи дискретного логарифмирования, которая является NP-полной.
* Обладает цифровой подписью, что позволяет проверить подлинность отправителя.

Минусы:

* Требует больших вычислительных мощностей при использовании больших ключей.
* Сложность выбора примитивного элемента поля может быть сложной.

Алгоритм кодирования и декодирования шифра RSA:

1. Генерация ключей:
   1. Выбирается два больших простых числа p и q.
   2. Вычисляется их произведение n = p \* q, которое будет использоваться в качестве модуля шифрования.
   3. Вычисляется значение функции Эйлера от числа n, φ(n) = (p-1)(q-1), которая является частью закрытого ключа.
   4. Генерируется открытый ключ (e, n), где e - целое число, взаимно простое с φ(n).
   5. Вычисляется значение секретного ключа d по формуле d ≡ e^(-1) (mod φ(n)).
2. Кодирование (шифрование):
   1. Представляется открытое сообщение M в виде числа m, где 0 <= m < n.
   2. Зашифрованное сообщение C вычисляется по формуле C ≡ m^e (mod n).
3. Декодирование (расшифрование):
   1. Зашифрованное сообщение C декодируется по формуле m ≡ C^d (mod n).

Плюсы RSA:

* Высокий уровень безопасности при использовании достаточно больших ключей.
* Возможность использования для шифрования/расшифрования больших объемов данных.
* Возможность использования для цифровой подписи.

Минусы RSA:

* Относительно медленная скорость выполнения операций шифрования и расшифрования по сравнению с некоторыми другими алгоритмами.
* Затраты памяти на хранение большого ключа и вычисления с большими числами.

Особенности RSA:

* При генерации ключей необходимо выбирать достаточно большие простые числа для обеспечения безопасности.
* На безопасность алгоритма можно повлиять с помощью атак перебора простых чисел или факторизации модуля n.

Алгоритм кодирования и декодирования шифра Вернама:

1. Генерация ключей:
   1. Генерируется случайная последовательность битов, размер которой равен размеру сообщения.
   2. Эта случайная последовательность является ключом шифрования/расшифрования.
2. Кодирование (шифрование):
   1. Исходное сообщение разбивается на биты.
   2. Каждый бит исходного сообщения M складывается с соответствующим битом ключа К.
   3. Полученный результат является зашифрованным сообщением C.
3. Декодирование (расшифрование):
   1. Зашифрованное сообщение C разбивается на биты.
   2. Каждый бит зашифрованного сообщения C складывается с соответствующим битом ключа К.
   3. Полученный результат является исходным сообщением M.

Плюсы шифра Вернама:

* Шифр является абсолютно надежным, если ключ генерируется правильно и используется только один раз.
* Математически не возможно взломать шифр без знания ключа.
* Позволяет зашифровывать и расшифровывать сообщения любой длины.

Минусы шифра Вернама:

* Массивный ключ, который должен быть длиной, равной длине сообщения, требует больших затрат памяти.
* Ключи должны быть действительно случайными и использоваться только