Криптосистема McEliece:

1. Случайная генерация ключа
2. Случайное шифрование
3. Детерминированное расшифрование

Работаем в поле Галуа (конечное поле).
В случае простого порядка – это просто кольцо вычетов.
В общем случае строится с помощью многочленов.

Генерация ключей (k < n)

S = случайная невырожденная матрица k x k
G = порождающая матрица размером k x n некоторого линейного кода
P = случайная матрица перестановки размером n x n
(перемешали строки единичной матрицы)
Закрытый ключ = (S, G, P)

G_ = S x G x P = матрица kxn
Знаем, что линейный код может исправлять t ошибок
Открытый ключ = (G_, t)

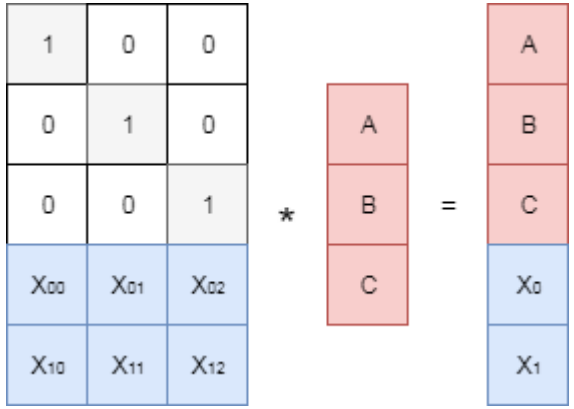
Шифрование

m = сообщение длины k
z = случайный вектор длины n, искажающий t символов в m
c = m x G_ + z = шифротекст длины n

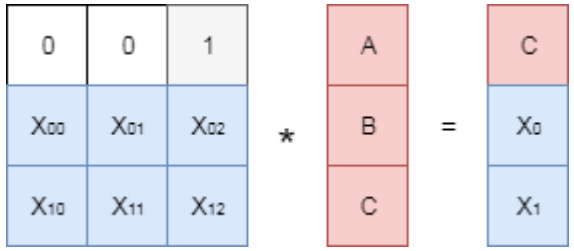
Расшифрование

c_ = c x P^-1 = m x S x G + z x P^-1
P = матрица перестановки => z x P^-1 содержит тоже t ошибок
m_ = декодирование c_ = m x S
m = m_ x S^-1

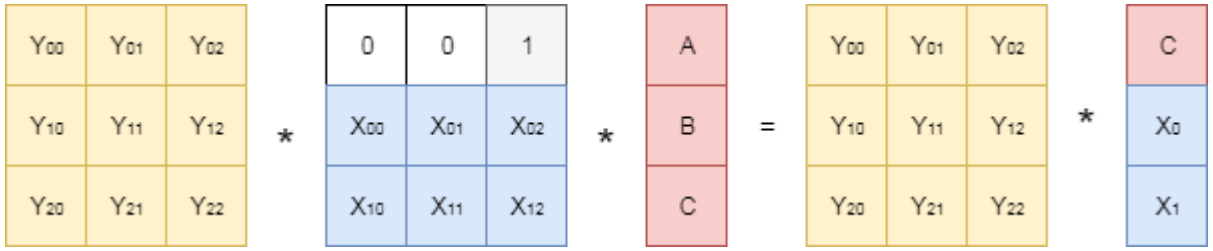
Код Рида-Соломона – основная идея



G x данные = избыточные данные

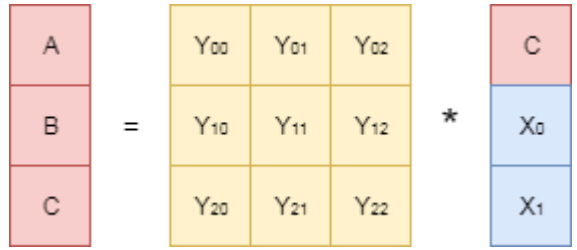


Потеряли A и B – часть исходных данных



Восстановили A и B

Y = матрица, обратная к обрезанной G
Таким образом, слева на самом деле стоит
E x данные = данные



Источник (теория про коды и про поля Галуа)
https://habr.com/ru/companies/yadro/articles/336286/
https://habr.com/ru/companies/yadro/articles/341506/

Задача с CTF

Дано: в качестве линейного кода используется алгоритм Рида-Соломона (можем вычислить G), в алгоритме шифрования не используется вектор z, известны матрицы G_ и P (n и k, соответственно тоже – это размер G_), порядок поля Галуа, а также зашифрованное сообщение c.
Найти: сообщение m. По сути: надо найти S, всё остальное уже есть.
Решение: G_ = S x G x P => G_ x P^-1 = S x G, причём из-за использования кода Рида-Соломона левая часть G – это единичная матрица. Убираем из матрицы G_ x P^-1 лишние правые столбцы и получаем S.

Реализация (python + numpy + galois)

Конструирование поля Галуа, конструирование и декодирование кода Рида-Соломона реализовано в библиотеке galois; это дополнение к библиотеке numpy, позволяющее numpy проводить многие операции в поле Галуа.

Сообщение (строка) переводится в байты, дополняется до длины k.

Многие функции уже были написаны в задании, надо было только получить матрицу S и дописать алгоритм расшифровки.

Не сразу было замечено, что левая часть матрицы G – это единичная матрица, поэтому совершается лишнее действие, а именно домножение обрезанной матрицы G x P^-1 на матрицу, обратную к обрезанной матрице G.

Мой репозиторий с кодом: https://github.com/vovuas2003/McEliece