Дедлайн: 12 мая

1. *(7 баллов)* Пусть $(\alpha_1, \ldots, \alpha_n)$ — набор различных ненулевых элементов конечного поля $\mathbb{F}_q$ (можно на выбор $q = 1024$ или $q = 1031$). Реализуйте

   (a) алгоритм кодирования сообщений $(m_0, \ldots, m_{k-1}) \in \mathbb{F}_q$ с помощью кода Рида–Соломона $RS_k(\alpha_1, \ldots, \alpha_n)$;

   (b) алгоритм генерации и добавления случайной ошибки $\mathbf{e}$ веса t к закодированному сообщению;

   На свой выбор реализуйте и протестируйте корректность работы любого из декодеров кодов Рида–Соломона *(например, декодер Велча–Берлекэмпа, декодер Шиозаки–Гао, декодер Берлекэмпа–Мэсси).*

   Рекомендуется использовать уже готовые реализации арифметики в конечных полях и арифметики многочленов (например, SageMath, SymPy, CoCoA, Wolfram Mathematica)

2. *(5 баллов)* Реализуйте и протестируйте протокол биометрической криптографии *Fuzzy Vault* на основе кодов Рида–Соломона, а именно функции LOCK и UNLOCK.

3. *(5 баллов)* Закодируйте кодом Рида–Маллера $RM(r = 2, m = 4)$ информационный вектор $\mathbf{m}$, внесите одну ошибку в получившееся кодовое слово, а затем пошагово его декодируйте.

   Замечание. *Пусть ваш месяц и день рождения записаны в виде* MMDD *(например, 0326, 26 марта). Тогда информационный вектор* $\mathbf{m}$ *следует считать равным двоичному представлению этой последовательности (без учёта незначащих нулей). Если в получившейся двоичной последовательности меньше* 11 *символов, добавьте необходимое количество нулей в начале слова. Пример: 26 марта* $\mapsto$ 0326 $\mapsto$ 00101000110.

   Бонусные баллы: 5 баллов первым 10 сдавшим, 3 балла сдавшим до 25 апреля. Также бонусные баллы можно получить за реализацию алгоритмов на компилируемых языках программирования (Rust, C++, C), а также за программную реализацию декодера кодов Рида–Маллера.

Algorithm 1: Декодер Велча–Берлекэмпа

Input: $(z_1, \ldots, z_n)$ — сообщение, принятое по каналу связи

1. Пусть $r = \lfloor \frac{n-k}{2} \rfloor$ — максимальный вес ошибки, которую можно исправить. Найти многочлены
$$L(x) = \sum_{i=0}^{r} L_i x^i, \quad N(x) = \sum_{i=0}^{r+k-1} N_i x^i,$$
которые удовлетворяющие следующей ОСЛУ
$$\begin{cases} N(\alpha_1) = z_1 L(\alpha_1) \\ N(\alpha_2) = z_2 L(\alpha_2) \\ \ldots \\ N(\alpha_n) = z_n L(\alpha_n) \end{cases}$$
2. $m(x) = N(x)/L(x)$ — исходное сообщение

Algorithm 2: Декодер Шиозаки–Гао

Input: $(z_1, \ldots, z_n)$ — сообщение, принятое по каналу связи

1. Построить с помощью интерполяции многочлен $T(x)$, $\deg(T) < n$, такой что $T(\alpha_j) = y_j$ для $1 \le j \le n$.
2. С помощью расширенного алгоритма Евклида для найти многочлены $L(X)$ и $N(x)$, что
$$\begin{cases} L(X)T(x) \equiv N(X) \pmod{W(x)}, \quad \text{где } W(x) = \prod_{i=1}^{n}(x - \alpha_i) \\ \deg N(x) \le r + k - 1, \\ \deg L(x) \to \max \end{cases}$$
3. $m(x) = N(x)/L(x)$ — исходное сообщение

Algorithm 3: LOCK

Input: $A = \{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_q^*$ — образец биометрии, $m(x)$ — секретный многочлен небольшой степени

```
S ← Dict();
foreach β ∈ F_q^* do
    if β ∈ A then
        S[β] ← m(β);
    else
        S[β] ← rand();
end
return S;
```