МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича»

Кафедра Защищенных систем связи

Дисциплина «Основы криптографии с открытыми ключами»

Лабораторная работа № 10-2

Исследование протокола скрытого определения местоположения точек интереса пользователя с учетом типа POI

Вариант 4

Выполнил: ст. г. ИКТЗ-83

Громов А. А.

Проверил: Яковлев В. А.

Санкт-Петербург
2021

Цель лабораторной работы

Закрепить теоретические знания студентов по разделу: “Гомоморфное шифрование”. Ознакомиться с протоколом скрытого определения точек интереса мобильного пользователя на основе изученных алгоритмов криптосистем Пэйе и Рабина.

Исходные данные

Выбор ячейки производится следующим образом:

координата $i = N\ mod 5 + 1$,

координата $j = (D + N)\ mod 5 + 1$,

где N – номер студента по журналу, D – день выполнения лабораторной работы.

Получаем:

$i = N\ mod\ 5 + 1 = 4\ mod\ 5 + 1 = \mathbf{5}$

$j = (D + N)\ mod\ 5 + 1 = (5 + 4)\ mod\ 5 + 1 = \mathbf{5}$

Ход работы

Генерация ключей

Выбираем два больших простых числа $p_1,\ q_1$, таких что $N_1 = p_1 q_1 > M$, где $M = \max(d_{i,j})$ – самое большое целое число из базы данных сервера, содержащей информацию о ближайших POIs. С учетом того, что $M = 14700$, $N_1 > 14700$. Числа $p_1,\ q_1$ выбрать из диапазона 122-160.

Также необходимо, чтобы все сгенерированные числа удовлетворяли условию: $G\ mod\ 4 = 3$, где G – сгенерированное простое число.

$p_1 = 139$

$q_1 = 151$

$N_1 = p_1 q_1 = 139 * 151 = 20989$

Выбираем следующие два больших простых числа $p_2,\ q_2$, так, чтобы ${N^2}_1 \cdot 100 < N_2 < {N^4}_1$, где $N_2 = p_2 q_2$. Числа $p_2,\ q_2$ выбрать, исходя из диапазона: $p_2 > \sqrt{{N^2}_1 \cdot 100},\ q_2 < \sqrt{{N^4}_1}$.

$p_2 = 215051$

$q_2 = 20945051$

$N_2 = p_2 q_2 = 215051 * 20945051 = 4504254162601$

```
(%i35) p1:139;

       power_mod(2,p1-1,p1);
       power_mod(3,p1-1,p1);
       power_mod(13,p1-1,p1);
       mod(p1, 4);
(%o31) 139
(%o32) 1
(%o33) 1
(%o34) 1
(%o35) 3

(%i40) q1: 151;

       power_mod(2,q1-1,q1);
       power_mod(3,q1-1,q1);
       power_mod(13,q1-1,q1);
       mod(q1, 4);
(%o36) 151
(%o37) 1
(%o38) 1
(%o39) 1
(%o40) 3
```

Рисунок 1. Проверка соответствия чисел $\boldsymbol{p_1}$, $\boldsymbol{q_1}$ заданным условиям.

```
(%i45) p2: 215051;

       power_mod(2,p2-1,p2);
       power_mod(3,p2-1,p2);
       power_mod(13,p2-1,p2);
       mod(p2, 4);
(%o41) 215051
(%o42) 1
(%o43) 1
(%o44) 1
(%o45) 3

(%i50) q2: 20945051;

       power_mod(2,q2-1,q2);
       power_mod(3,q2-1,q2);
       power_mod(13,q2-1,q2);
       mod(q2,4);
(%o46) 20945051
(%o47) 1
(%o48) 1
(%o49) 1
(%o50) 3
```

Рисунок 2. Проверка соответствия чисел $\boldsymbol{p_2}$, $\boldsymbol{q_2}$ заданным условиям.

Генерируем числа g_1 из множества $Z^*_{N_1{}^2}$ и g_2 из множества $Z^*_{N_2{}^2}$, удовлетворяющие условию:

$$\gcd\left(\frac{g^{\lambda} mod N^2 - 1}{N}, N\right) = 1.$$

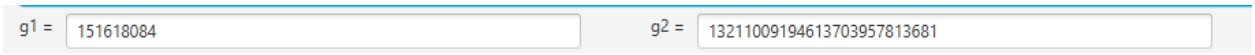

Рисунок 3. Генерация чисел $\boldsymbol{g_1}$, $\boldsymbol{g_2}$.

$g_1 = 151618084$

$g_2 = 132110091946137039578136 81$

```
(%i91) g:151618084;
       p:139;
       q:151;
       N:p·q;
       y:lcm(p−1, q−1);
       N_2:N^2;
       a:power_mod(g, y, N_2)−1;
       gcd(a/N, N);
(%o84) 151618084
(%o85) 139
(%o86) 151
(%o87) 20989
(%o88) 3450
(%o89) 440538121
(%o90) 439404715
(%o91) 1
```

```
(%i99) g:132110091946137039578136 81;
       p:215051;
       q:20945051;
       N:p·q;
       y:lcm(p−1, q−1);
       N_2:N^2;
       a:power_mod(g, y, N_2)−1;
       gcd(a/N, N);
(%o92) 132110091946137039578136 81
(%o93) 215051
(%o94) 20945051
(%o95) 4504254162601
(%o96) 90084660050
(%o97) 20288305561308435747085201
(%o98) 15394068446706222747354238
(%o99) 1
```

Рисунок 4. Проверка соответствия чисел $\boldsymbol{g_1}$, $\boldsymbol{g_2}$ заданным условиям.

Таким образом, имеем следующие ключи:

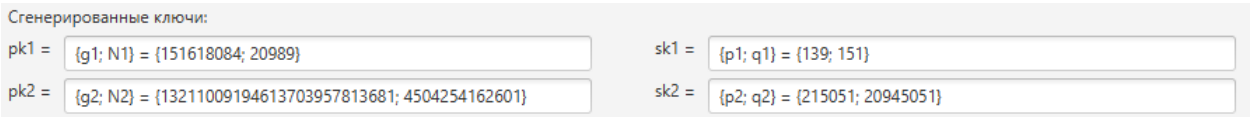

Рисунок 5. Сгенерированные ключи.

Формирование запроса

1. Шифрование POI типа t на первом открытом ключе

Для каждого $l \in \{1, 2, \dots, m\}$ пользователь выбирает случайное целое число $r_l \in Z^*_{N_1}$ и вычисляет криптограммы c_l:

$$c_l = \begin{cases} Encrypt\ (1, pk_1) = {g_1}^1 {r_l}^{N_1} \left(mod\ {N_1}^2\right) \text{ если } l = t \\ Encrypt\ (0, pk_1) = {g_1}^0 {r_l}^{N_1} \left(mod\ {N_1}^2\right) \text{ если } l \neq t \end{cases},$$

где t – тип точек интереса, про который пользователь запрашивает информацию.

Вычисляем криптограммы c_l с при $t = 1$:

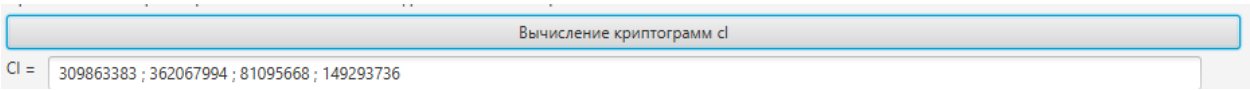

Рисунок 6. Вычисление криптограмм $\boldsymbol{c_l}$.

2. Шифрование координаты i своей ячейки на втором открытом ключе.

Для каждого $l' \in \{1, 2, \dots, n\}$ пользователь выбирает случайное целое число $r_{l'} \in Z^*_{N_2}$ и вычисляет криптограммы $c'_{l'}$:

$$c'_l = \begin{cases} Encrypt\ (1, pk_2) = {g_2}^1 {r'_{l'}}^{N_2} \left(mod\ {N_2}^2\right) \text{ если } l = i \\ Encrypt\ (0, pk_2) = {g_2}^0 {r'_{l'}}^{N_2} \left(mod\ {N_2}^2\right) \text{ если } l \neq i \end{cases},$$

где i – первая координата ячейки, в которой находится пользователь.

Вычисляем криптограммы c'_l:

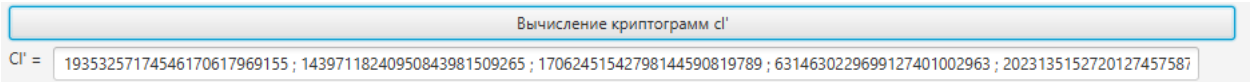

Рисунок 7. Вычисление криптограмм $\boldsymbol{c'_l}$.

3. Шифрование координаты j своей ячейки на втором открытом ключе

Далее пользователь выбирает случайное целое число $r \in Z^*_{N_2}$ и вычисляет еще одну криптограмму c:

$$c = Encrypt(j, pk_2) = {g_2}^j r^{N_2}\ \left(mod\ {N_2}^2\right),$$

где j – вторая координата ячейки, в которой находится пользователь.

Вычисляем криптограмму c:

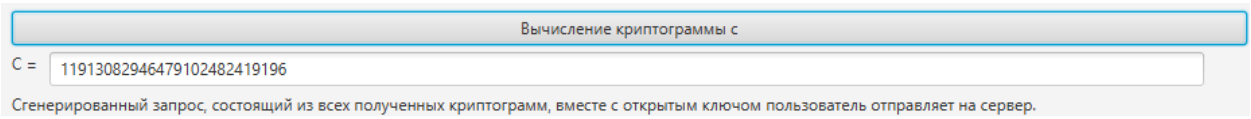

Рисунок 8. Вычисление криптограммы $\boldsymbol{c}$.

Все полученные криптограммы пользователь отправляет на сервер в качестве запроса.

Генерация ответа сервера

1. Шифрование значений POIs для всех ячеек на первом открытом ключе.

Вычисляется $C_{\alpha,\beta}$, где $\alpha \in \{1, 2, \dots, n\}$, $\beta \in \{1, 2, \dots, n\}$:

$$C_{\alpha,\beta} = \prod_{l=1}^{m} {c_l}^{{d_{\alpha,\beta,l}}^2} (mod\ {N_1}^2)\ .$$

2. Вторичное шифрование POIs и дополнительное шифрование для координаты j на втором открытом ключе.

Для каждого $\beta = \{1,2, \dots, n\}$ выбирается ω_β – целое число из множества ${Z^*}_{N_2}$ и вычисляется $R = \{C_1, C_2, \dots, C_n\}$:

$$C_\beta = \left(\frac{c}{g^\beta}\right)^{\omega_\beta} \prod_{\alpha=1}^{n} {c'}_\alpha^{{C_{\alpha,\beta}}^2}\ (mod\ {N_2}^2).$$

Вычисление криптограмм R:

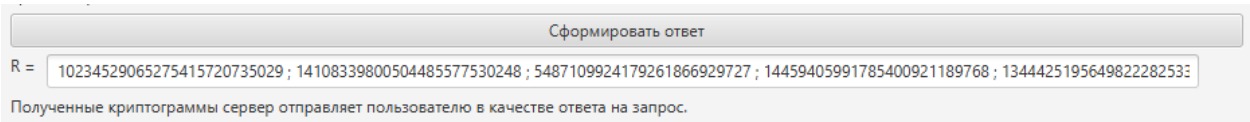

Рисунок 9. Вычисление криптограмм ***R***.

После того, как сервер вычислил криптограммы R, он посылает их пользователю в качестве ответа на полученный запрос.

Получение ответа

Пользователь получает ответ от сервера в виде $R = \{C_1, C_2, \dots, C_n\}$:

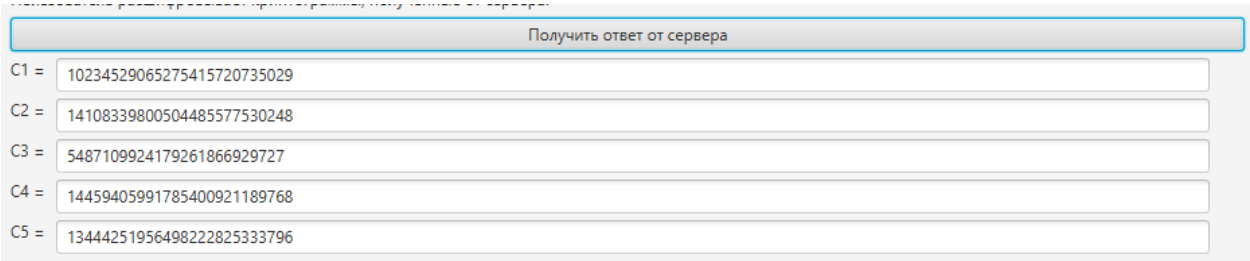

Рисунок 10. Получение пользователем криптограмм ***R***.

Далее пользователь выбирает только то значение, порядковый номер которого соответствует второй координате j его местоположения и выполняет расшифровку данных, состоящую из четырех шагов.

В нашем случае выбираем криптограмму C_5.

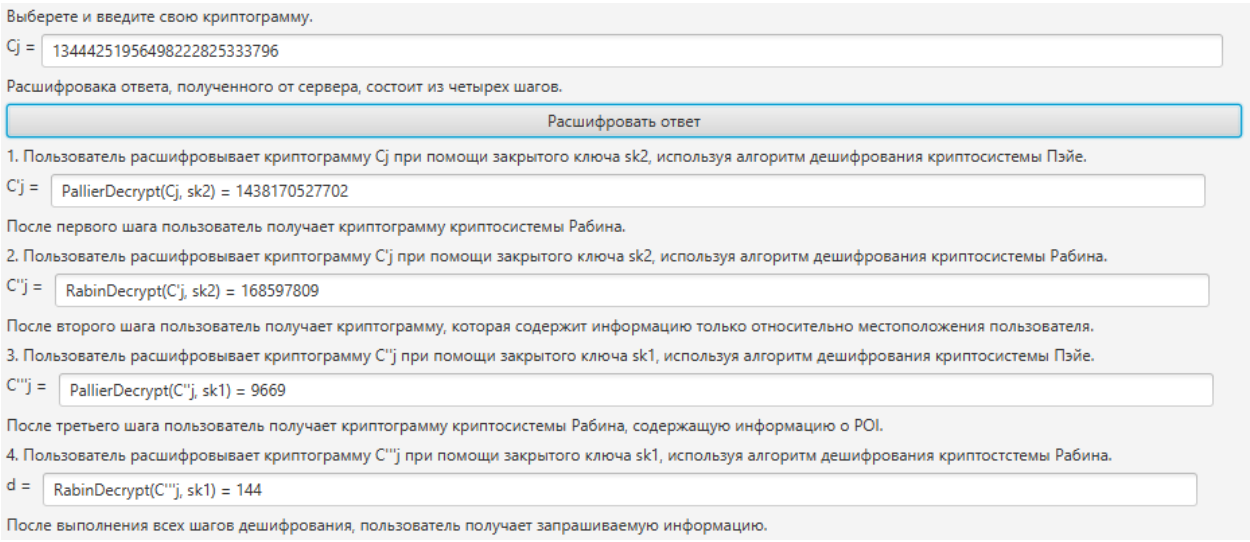

Рисунок 11. Выполнение алгоритма дешифрования.

В результате четвертого шага дешифровки, пользователь получает информацию о ближайшей точке интереса типа t для своей ячейки (i, j), представленную в десятичном виде.

Для вычисления координат и типа POI переводим d в двоичную форму (длиной 8 бит):

$d = 144 = 10010000_2$

Первые три бита содержат первую координату, следующие три бита – вторую координату, последние два бита – тип точки интереса. При этом к значениям каждой группы битов нужно добавить 1. Таки образом видим, что ближайший банкомат (t = 1) находится в ячейке: {5; 5}.

Видим, что полученное значение совпадает с данными на карте:

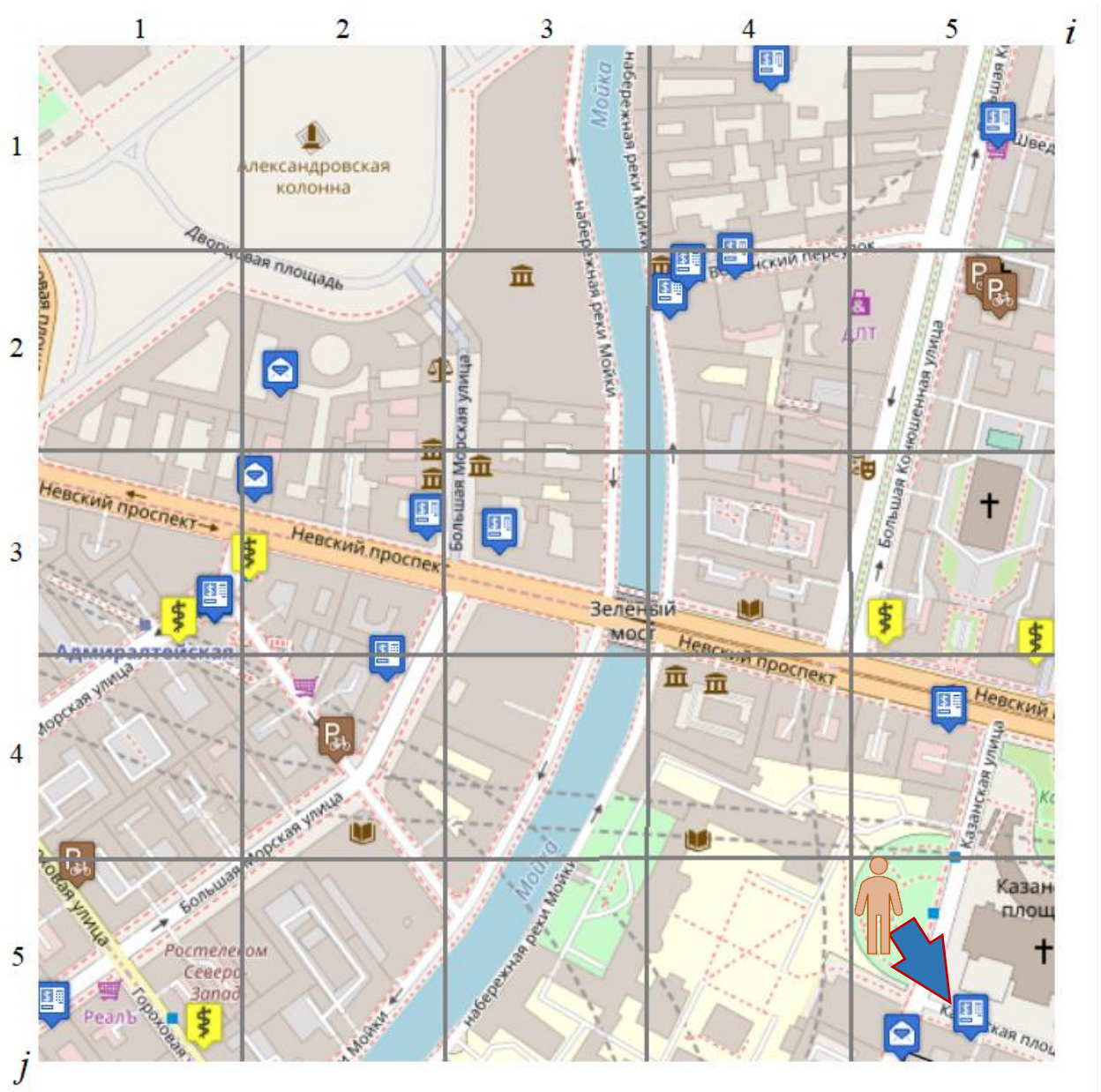

Рисунок 12. Проверка по карте.

Попробуем расшифровать криптограмму полученную от сервера, порядковый номер которой не равен второй координате $j = 5$:

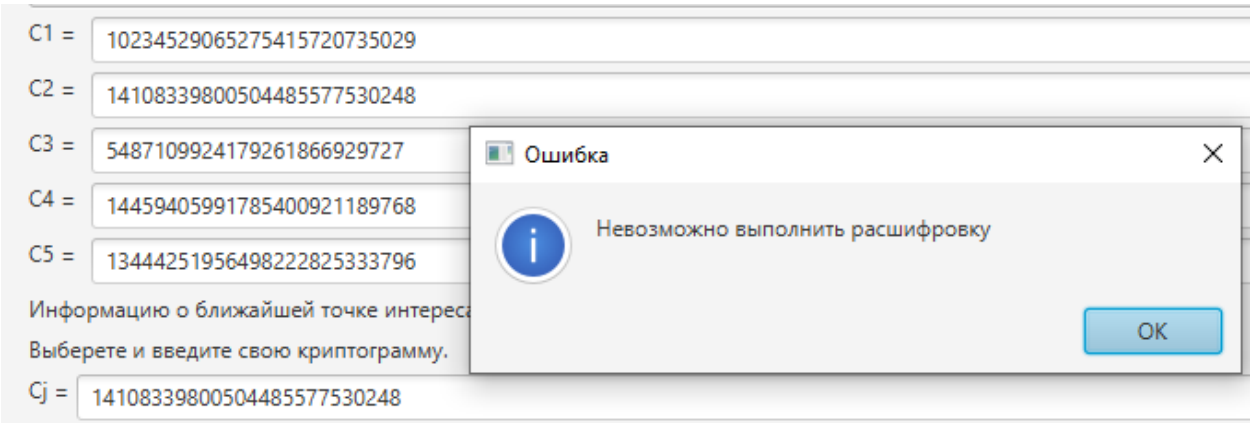

Рисунок 13. Попытка расшифровки криптограммы с индексом $\boldsymbol{l \neq j}$.

Видим, что благодаря тому, что для формирования ответа сервер использует шифрование Рабина и Пэйе, пользователь не может расшифровать данные ни для какой ячейки, кроме своей.

Повторим процедуру скрытого определения POIs для остальных типов POIs (при использовании тех же ключевых данных). Получаем следующие результаты:

Ближайшая велопарковка: (5, 2)

Ближайшая аптека: (5, 4)

Ближайшее отделение почты: (5, 5)

Видим, что полученные значения также совпадают с данными на карте:

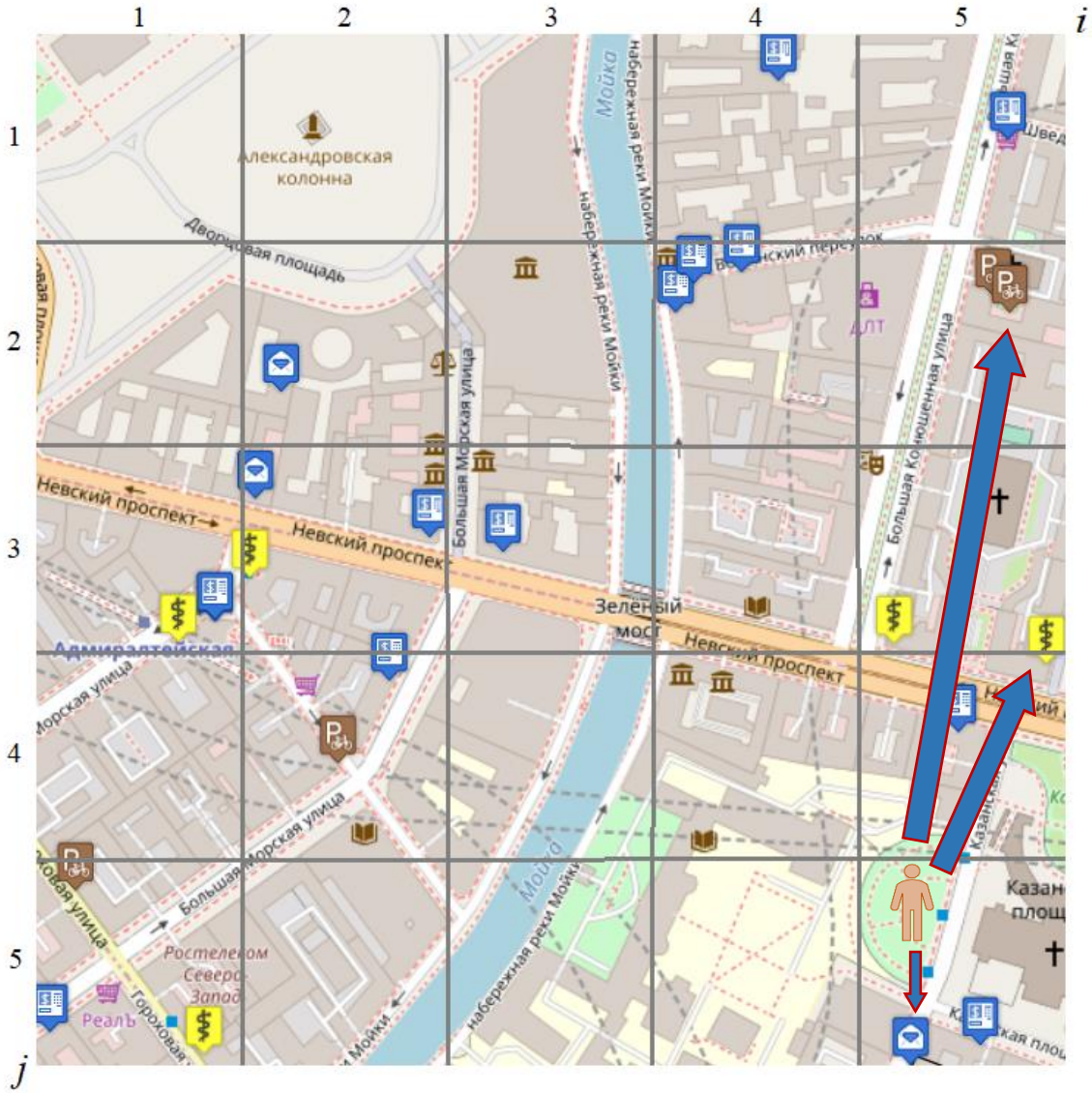

Рисунок 14. Проверка по карте.

Вывод:

В ходе выполнения данной лабораторной работы были закреплены теоретические знания по разделу «Гомоморфное шифрование», произведено ознакомление с протоколом скрытого определения точек интереса мобильного пользователя на основе алгоритмов КС Пэйе и Рабина.