Перед нами сервер, через который можно зашифровать сообщения причём собственным ключом. Для шифрования используется шифр Эль-Гамаля, внутри которого применяется псевдослучайный генератор, от доверенного центра. Также можем увидеть, что псевдослучайный генератор приставляет из себя линейный регистр сдвига с обратной связью.

Оценка уязвимостей и ситуаций решения:

1. Можно давать собственный открытый ключ. А значит можно дать и плохие не подходящие под корректную работу числа. При этом проверка идёт только на простоту подаваемого числа.
2. Шифр Эль-Гамаля полностью зависит от случайных чисел, и если знать их, то можно расшифровать сообщение.
3. Линейный регистр сдвига – это плохой псевдослучайный генератор, который эффективно восстанавливается с помощью алгоритма Берлекэмпа — Мэсси

Оценив эти факты, можно прийти к следующему способу атаки:

1. Шифруем любое сообщение с помощью шифра Эль-Гамаля используя не сильно простое число, то есть простое число p такое что , где – это маленькие простые числа. Тогда проблему дискретного логарифмирование можно будет решить с помощью Алгоритм Полига-Хеллмана. К примеру можно использовать следующее простое число p=0x1d65290b673d90c1c4ff309b7ebed2d3f277c9d1799f33219bdc36cbff6e8ce6afc9286e442befdb2e72d0fd2db4b02b785a997f9e70154be933946da2bf0dfd3189a2ccdfc05f5d5709b78701b5ebd23039041d7ed40653b83792f3c5ce592a80b3318ffad786e74e69aea06e27d4f9ce96a9ce0973b5160c961e62d7c8681a87, так как p-1 = 2 · 33487 · 33827 · 34487 · 34687 · 35279 · 36067 · 36277 · 36293 · 37409 · 37691 · 37831 · 38149 · 38261 · 39293 · 40213 · 40423 · 40697 · 42403 · 42533 · 42737 · 43543 · 44257 · 45281 · 46439 · 47237 · 47317 · 47407 · 47569 · 48073 · 48487 · 48947 · 49253 · 49459 · 49597 · 50287 · 50873 · 51473 · 51749 · 52177 · 52237 · 53681 · 53813 · 54251 · 54751 · 54917 · 55061 · 55229 · 55511 · 55541 · 56197 · 57457 · 58537 · 60317 · 62497 · 62617 · 62633 · 63113 · 63463 · 63607 · 64007 · 64091 · 64123 · 64151 · 64271 · 64301 · 65053. То есть раскладывается на маленькие простые числа. В таком случае зная, что можно найти k.

Алгоритм Полига-Хеллмана:

Вход: , g, p,

Выход: k

1. Для всех i от 1 до n:
   1. ,
   2. , что в данном случае можно сделать перебором, так как степень по модулю p равно , а это небольшое число
2. k находится с помощью китайской теореме об остатках
3. Несколько раз применяя алгоритм Полига-Хеллмана можно найти несколько значений псевдослучайного генератора. Мы не знаем сколько нам понадобиться для восстановления ключа генератора, но можно взять с запасом. Вообще в данном случае потребуется минимум 8 подряд идущих значений k. Тогда имея последовательность бит линейного регистра сдвига с обратной связью можно восстановить ключ с помощью алгоритма Берлекэмпа — Мэсси.

Алгоритм Берлекэмпа — Мэсси:

Пусть дана последовательность . Тогда

1. C(D)=1, L=0, m=-1, B(D)=1, N=0.

2. Пока N<n

2.1

2.2 Если d=1

2.2.1 T(D)=C(D),

2.2.2 Если L≤0,5N, то L=N+1-L, m=N, B(D)=T(D)

2.3 N=N+1

3. Выход: L, C(D)

1. Узнав ключ псевдослучайного генератора, можно сгенерировать все следующие псевдослучайные числа и узнать все предыдущие при этом расшифровав все зашифрованные сообщения. Так если мне известно случайное число k в шифре Эль-Гамалаля, то можно получить m из из этого можно найти , так как нам известны открытые ключи и зашифрованные сообщения а также само значение k.

Устранить эту уязвимость можно включив в проверку числа на простоту также проверку на его сильную простоту. То есть что p-1 должно состоять только из больших простых чисел, а желательно вообще выглядеть как 2\*q где q это тоже простое число