Хотелось бы, чтобы задача (может быть, не только эта, но и весь раздел нерешенных задач) развивалась по нескольким направлениям:

Алгоритмическое – предложить новый алгоритм решения задачи либо оптимизировать существующий. Кратко опишу известные мне:

1. На основе таблицы add-xor разностей подстановки g. Таблица рассчитывается для разностей двух пар "открытый текст/шифр текст", клетка таблицы содержит список допустимых ключей для соответствующей разности. На этапе дешифрования множество ключей есть пересечение множеств допустимых ключей для каждой наблюдаемой разности.
2. Путем составления системы уравнений, описывающей целевую функцию. Обычно, такое удается сделать над двоичным полем. Для каждой наблюдаемой пары в систему добавляется экземпляр системы, но с известными входными и выходными значениями и общими ключевыми переменными. Решать систему можно, например, с помощью базисов Гребнера.
3. Поиск фиксированной точки. Имея две наблюдаемые пары, можно (функционально) выразить ключевые переменные через себя, а также через известные значения блоков открытых/шифр текстов.
4. Перебор. Попробуем провести мысленный эксперимент, который заключается в следующем. Составим систему из алгоритма 2, занумеруем биты ключа и будем их перебирать по порядку, добавляя текущее значение в систему. Если найдено противоречие (1 = 0), то пробуем другое значение текущего бита ключа либо предыдущего бита и т.д., если противоречия нет, то переходим к следующему биту. Т.о. перебор представляет собой поиск в глубину в дереве возможных ключей.

Аналитическое – провести анализ алгоритма, определить его сложность, выявить его слабые места, каким образом можно усложнить/сделать легче задачу, чтобы алгоритм был неэффективным/эффективным. Например:

1. Алгоритм 1 представляется наиболее эффективным, но требует на этапе предвычислений огромных затрат памяти и времени. Даже криптоаналитик с его большим потенциалом не сможет его реализовать. Значит, требуется (обязательно) знать структуру подстановки g?
2. Алгоритм 2 поэтому кажется более привлекательным с практической точки зрения. Однако теперь вопрос в решении системы. Существуют алгоритмы нахождения базиса Гребнера (например, алгоритм Бухбергера), однако они все являются экспоненциальным (если не ошибаюсь). Как решить систему?
3. Алгоритм 3 в некотором роде является обобщением алгоритма 1 на случай более сложного описания целевой функции (например, большее число ключевых переменных и большее число вызовов подстановки g). Однако, мне не известны алгоритмы нахождения фиксированных точек. Более того, чтобы алгоритм был эффективным необходимо знание структуры подстановки g. Как найти фиксированную точку?
4. Насчет алгоритма 4. Т.к. размер ключей в два раза больше размера текстов, то в среднем существует менее двух (если не ошибаюсь) подходящих ключей для двух и более пар. Понятно, что запреты будут возникать почти для всех ключей. Вопрос в том, сколько в среднем бит ключа и в каком порядке потребуется перебрать, чтобы получить противоречие, если наблюдается T пар (и есть ли вообще какая либо зависимость)? (Если для любых двух пар существует единственный допустимый ключ, то для двух и более пар потребуется только один (неверный) бит, чтобы получить противоречие). Каким образом можно эффективно определить, есть ли противоречие в системе или нет?
5. Алгоритмы 2 и 4 естественным образом расширяется на случай нескольких наблюдаемых пар. Алгоритмы 1 и 3 – только косвенным образом - путем пересечения множеств ключей

для каждых двух пар "открытый/шифр текст". Насколько важным является это свойство? (Опыт подсказывает, что достаточно важным).

6. Сложность представленных алгоритмов – экспоненциальная. Для более точной оценки требуется реализация (или хотя бы более формальное описание) алгоритмов. Однако, все алгоритмы упрощаются, если предположить, что подстановка g имеет блочную структуру.