**Лекция 14. Интерполяционный криптоанализ**. Ключ восстонавления. Использование метода.

*Метод интерполяции* был предложен в 1997 г. датскими криптологами Т.Якобсеном и Л. Кнудсеном и описан в их работе. Атаке подвержены алгоритмы шифрования, использующие достаточно простые алгебраические операции, в результате чего криптоаналитик может построить некий полином, определяющий соотношение между шифртекстом и открытым текстом.

Авторы атаки доказали, что если число ненулевых коэффициентов полинома (обозначим его *п)* не превышает 2m, где *m* — размер блока шифруемых данных в битах, то возможна атака на этот алгоритм 34 шифрования, которая находит алгоритм, эквивалентный расшифровыванию (или зашифровыванию) данных искомым секретным ключом. Для такой атаки требуется выполнение п тестовых операций шифрования при наличии п известных открытых текстов.

В этой атаке, алгебраическая функция используется для представления [S-Box](https://ru.wikipedia.org/wiki/S-%D0%B1%D0%BB%D0%BE%D0%BA%D0%B8). Это может быть простое [квадратичная функция](https://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B0%D0%B4%D1%80%D0%B0%D1%82%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F), [полином](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BD%D0%BE%D0%BC) или [рациональная функция](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) над [полем Галуа](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B5_%D0%93%D0%B0%D0%BB%D1%83%D0%B0). Еe коэффициенты могут быть определены с помощью стандартных методов [интерполяции Лагранжа](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%D1%87%D0%BB%D0%B5%D0%BD_%D0%9B%D0%B0%D0%B3%D1%80%D0%B0%D0%BD%D0%B6%D0%B0), с использованием известных открытых текстов как точек данных. Кроме того, выбранные [открытыe тексты](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D1%8B%D0%B9_%D1%82%D0%B5%D0%BA%D1%81%D1%82) могут быть использованы для упрощения уравнений и оптимизировать атаку. В простейшем варианте интерполяционная атака выражает зашифрованный текст в виде многочлена от текста. Если многочлен имеет относительно низкое число неизвестных коэффициентов, то с набором пар открытого текста / зашифрованного текста , полином может быть восстановлен. Зная полиномом восстановления, атакующий имеет представление о шифровании без точного знания секретного [ключа](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_%28%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F%29). Интерполяционная атака невозможна, если использовать не [непрерывную функцию](https://ru.wikipedia.org/wiki/%D0%9D%D0%B5%D0%BF%D1%80%D0%B5%D1%80%D1%8B%D0%B2%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F).

## **Пример.**

Пусть [итерация](https://ru.wikipedia.org/wiki/%D0%98%D1%82%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D1%8F) шифрования задаётся как



где c 0 {\displaystyle c\_{0}} - это открытый текст, k i ∈ K {\displaystyle k\_{i}\in K} - секретный раундовый ключ, c r {\displaystyle c\_{r}} -зашифрованный текст для r {\displaystyle r} r- раундовой итерации шифрования.

Рассмотрим 2-раундовый шифр. Пусть x {\displaystyle x} x- сообщение и c {\displaystyle c} c- зашифрованный текст, тогда на выходе из 1-го раунда получаем

c 1 = ( x + k 1 ) 3 = ( x 2 + k 1 2 ) ( x + k 1 ) = x 3 + k 1 2 x + x 2 k 1 + k 1 3 , {\displaystyle c\_{1}=(x+k\_{1})^{3}=(x^{2}+k\_{1}^{2})(x+k\_{1})=x^{3}+k\_{1}^{2}x+x^{2}k\_{1}+k\_{1}^{3},} 

А на выходе из 2-го раунда:

c 2 = c = ( c 1 + k 2 ) 3 = ( x 3 + k 1 2 x + x 2 k 1 + k 1 3 + k 2 ) 3 {\displaystyle c\_{2}=c=(c\_{1}+k\_{2})^{3}=(x^{3}+k\_{1}^{2}x+x^{2}k\_{1}+k\_{1}^{3}+k\_{2})^{3}} 

Зашифрованный текст в виде полинома открытого текста выходов

p ( x ) = a 1 x 9 + a 2 x 8 + a 3 x 6 + a 4 x 4 + a 5 x 3 + a 6 x 2 + a 7 x + a 8 , {\displaystyle p(x)=a\_{1}x^{9}+a\_{2}x^{8}+a\_{3}x^{6}+a\_{4}x^{4}+a\_{5}x^{3}+a\_{6}x^{2}+a\_{7}x+a\_{8},} 

где a i {\displaystyle a\_{i}} - ключ, зависящий от [константы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D0%B0%D0%BD%D1%82%D0%B0)

Если мы будем использовать столько пар открытый текста/ зашифрованный текст сколько неизвестных коэффициентов в многочлене p ( x ) {\displaystyle p(x)} , то мы сможем построить многочлен. Это можно сделать, например, путём интерполяции Лагранжа. Когда неизвестные коэффициенты станут определены, мы будем иметь представление шифрования p ( x ) {\displaystyle p(x)} , без знания секретного ключа K {\displaystyle K} .

## **Интерполяционная атака Meet-In-The-Middle** (метод согласования)

Зачастую именно этот метод является эффективным.

Пусть есть r {\displaystyle r}  раундовый шифр с длиной блока m {\displaystyle m} , пусть z {\displaystyle z} z – выход шифра после s {\displaystyle s} s раундов s < r {\displaystyle s<r} . Мы выразим значение z {\displaystyle z} z в виде полинома открытого текста x {\displaystyle x} , и как многочлен зашифрованного текста c {\displaystyle c} . Пусть g ( x ) ∈ G F ( 2 m ) [ x ] {\displaystyle g(x)\in GF(2^{m})[x]}  выражает z {\displaystyle z} через x {\displaystyle x} , и пусть h ( c ) ∈ G F ( 2 m ) [ c ] {\displaystyle h(c)\in GF(2^{m})[c]}  выражает z {\displaystyle z} через c {\displaystyle c} . Полином g ( x ) {\displaystyle g(x)} мы получим вычисляя вперед и не используя повторных формул шифра до раунда, включая s {\displaystyle s} . Полином h ( c ) {\displaystyle h(c)}  мы получим вычисляя назад и используя повторную формулу шифра до s + 1 {\displaystyle s+1}  раунда.

Таким образом получается, что 

и если оба многочлена g {\displaystyle g}  и h {\displaystyle h}  с небольшим количеством коэффициентов, то мы можем решить уравнение для неизвестных коэффициентов.

## **Ключ восстановления**

Если убрать последний раунд r {\displaystyle r} - раундового повторного шифра длиной блока m {\displaystyle m} m, выход шифра становится y ~ = c r − 1 {\displaystyle {\tilde {y}}=c\_{r-1}} . Получается шифр со сниженной сложностью шифрования. Идея состоит в том, чтобы сделать предположение на последнем  k r {\displaystyle k\_{r}} раундового ключа мы можем расшифровать один раунд, чтобы получить выход  y ~ {\displaystyle {\tilde {y}}} приведённого шифра. Тогда, чтобы проверить предположение, нужно использовать интерполяционную атаку на сниженный шифр либо обычным способом, либо с помощью метода Meet-In-The-Middle.

Обычным способом мы выражаем вывод  y ~ {\displaystyle {\tilde {y}}} приведённого шифра в виде полинома открытого текста x {\displaystyle x} . Получается многочлен p ( x ) ∈ G F ( 2 m ) [ x ] {\displaystyle p(x)\in GF(2^{m})[x]} . Тогда, если мы можем выразить p ( x ) {\displaystyle p(x)}  с  n {\displaystyle n} коэффициентами, тогда с помощью n {\displaystyle n} известными различиями пары p / c {\displaystyle p/c} , мы можем построить многочлен. Чтобы проверить догадку на последнем раунде ключа, надо проверить с одной дополнительной парой p / c {\displaystyle p/c} , если он считает, что 

то с большой долей вероятности предположение последнего раунда ключ был правильным. Если нет, то нужно сделать ещё одно предположение ключа.

Методом Meet-In-The-Middle мы выражаем выход z {\displaystyle z} из раунда s < r {\displaystyle s<r}  в виде полинома открытого текста x {\displaystyle x}  и в виде полинома выхода пониженного шифра y ~ {\displaystyle {\tilde {y}}}  Пусть многочлены будут выражены через p {\displaystyle p}  и q {\displaystyle q}  коэффициенты, соответственно. Тогда с q + p − 2 {\displaystyle q+p-2}  известным различием пар  p / c {\displaystyle p/c} мы можем найти коэффициенты. Чтобы проверить догадку на последнем раунде ключа, проверяем с одной дополнительной парой p / c {\displaystyle p/c} , если равенство  выполняется, то с большой долей вероятности предположение ключа последнего раунда был правильными. Если нет, то делаем ещё предположение ключа.

После того, как мы нашли правильный ключ последнего раунда, можно продолжать аналогичным образом на остальных раундовых ключах.

## **Использование**

Атака Meet-in-the-Middle может быть использована в варианте для атаки S-блоков, которые используют [обратную функцию](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%80%D0%B0%D1%82%D0%BD%D0%B0%D1%8F_%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F), потому что с m {\displaystyle m} -битной S-блоком,  S : f ( x ) = x − 1 = x 2 m − 2 {\displaystyle S:f(x)=x^{-1}=x^{2^{m}-2}} в G F ( 2 m ) {\displaystyle GF(2^{m})} . Блочный шифр SHARK использует [SP-сеть](https://ru.wikipedia.org/wiki/SP-%D1%81%D0%B5%D1%82%D1%8C) с S-блоками S : f ( x ) = x − 1 {\displaystyle S:f(x)=x^{-1}} . Шифр устойчив к дифференциальному и линейному криптоанализу после небольшого количества раундов. Однако он был сломан в 1996 году Томасом Якобсен и Ларс Кнудсен, которые использовали интерполяционную атаку. Обозначим через SHARK ( n , m , r ) {\displaystyle (n,m,r)} версию SHARK с размером блока  n m {\displaystyle nm} бит с помощью  n {\displaystyle n} параллельных m {\displaystyle m} -битных S-блоков с количеством раундов r {\displaystyle r} . Якобсен и Кнудсен обнаружили, что существуют интерполяционная атака на SHARK ( 8 , 8 , 4 ) {\displaystyle (8,8,4)} (64-битный блок шифрования), используя около  2 21 {\displaystyle 2^{21}} выбранных открытых текстов и интерполяционная атака на ( 8 , 16 , 7 ) {\displaystyle (8,16,7)}SHARK (128-битный блочный шифр), используя около  2 61 {\displaystyle 2^{61}} выбранных открытых текстов.

Также Томас Якобсен представил вероятностный вариант интерполяционной атаки с использованием алгоритма Мадху Судана для улучшения декодирования [кодов Рида-Соломона](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%B4_%D0%A0%D0%B8%D0%B4%D0%B0_%E2%80%94_%D0%A1%D0%BE%D0%BB%D0%BE%D0%BC%D0%BE%D0%BD%D0%B0). Эта атака может работать, даже если алгебраическое отношение между открытыми текстами и шифротекстами имеет лишь часть значений