Лекция 1

Односторонние функции

(Конспект: С. Исакова)

Основной задачей сложностной криптографии является построение математического аппарата, с помощью которого можно было бы доказывать утверждения о криптографической надежности. Во всех задачах, которые мы будем рассматривать, криптографическим противником будет выступать вероятностный полиномиальный по времени алгоритм. Вполне возможно, что это допущение не является правильным, и в реальности противником может быть человек, работа которого необязательно заключается в выполнении какого-то алгоритма.

К сожалению, основная задача сложностной криптографии на сегодняшний день не выполнена. Ни для каких интересных протоколов неизвестно безусловного доказательства их надежности. Современная сложностная криптография основана на некоторых предположениях. Эти предположения обычно утверждают существование некоторых объектов, кирпичиков криптографии, так называемых криптографических примитивов. Все утверждения о надежности выглядят следующим образом: если существует такой-то криптографический приметив, то данный протокол является надежным.

Одним их таких криптографических примитивов является односторонняя функция. Неформально говоря, односторонняя функция — такая функция, которую вычислить просто, а обратить трудно. Самое простое применение односторонней функции: это способ хранения паролей на почтовом сервере. Почтовый сервер хранит не сам пароль, а только результат применения односторонней функции к нему. Таким образом, даже если и файл, где хранится информация о паролях будет утерян, то сами пароли восстановить будет трудно.

Первая часть курса будет посвящена односторонним функциям, криптографическим протоколам и другим криптографическим примитивам, существование которых эквивалентно существованию односторонних функций.

1.1 Односторонние в наихудших случаях функции

В этом параграфе мы рассмотрим определение односторонней функции, основанное на классической теории сложности вычислений.

Определение 1.1. Функция f называется *честной*, если

$$\exists \text{ полином } p(n) : \ \forall x \ |x| \leq p(|f(|x|)|),$$

т.е. если функция не очень сильно сокращает вход.

Если функция f не является честной, то обратить f за полиномиальное время принципиально невозможно, так как размер результата не ограничен полиномом от размера входа.

Определение 1.2. Функция $f\colon \{0,1\}^* \to \{0,1\}^*$ называется *односторонней в наихудшем случае*, если:

- f вычислима за полиномиальное время,
- f честная,
- f^{-1} не вычислима (f не обратима) за полиномиальное время.

В дальнейшем все рассматриваемые односторонние функции и прочие конструкции, основанные на них, - по существу, честные (даже если определения допускают иное).

Теорема 1.1. *Односторонние в наихудшем случае функции существуют тогда и только тогда, когда* $\mathbf{P} \neq \mathbf{NP}$.

Доказательство. "$\Rightarrow$". Заметим, что задача обращения полиномиально вычислимой функции всегда принадлежит классу **NP**. Подсказкой в данной задаче будет аргумент функции. Таким образом, если $\mathbf{P} = \mathbf{NP}$, то мы можем обратить любую функцию, и односторонних в наихудшем случае функций не существует.

"$\Leftarrow$". Пусть $\mathbf{P} \neq \mathbf{NP}$. Построим одностороннюю в наихудшем случае функцию на основе задачи выполнимости. Пусть ϕ — формула в КНФ, а A — набор значений переменных. Определим функцию:

$$f(\phi, A) = \begin{cases} (\phi, 1^{|A|}), & \text{если } A \text{ выполняет } \phi, \\ (\phi, 0^{|A|}), & \text{если } A \text{ не выполняет } \phi. \end{cases}$$

Если бы мы могли обратить эту функцию, то мы могли бы найти выполняющий набор для любой выполнимой формулы, вычислив

$$f^{-1}(\phi, 1^{\text{кол-во переменных}}).$$

Таким образом, мы бы научились быстро решать задачу выполнимости. Но мы знаем, что эта задача NP-трудна. Противоречие. □

Почему это определение односторонней функции именно в наихудшем случае? Предположим, мы построили такую функцию, которую почти всегда можно быстро обратить (за исключением лишь небольшого числа входов или длин входов), или которую можно быстро обратить, если пользоваться случайными числами. Тогда такая функция будет односторонней согласно введенному определению, но при этом пользы от нее не будет (протокол не является надежным, если его почти всегда можно взломать). Поэтому, принято рассматривать несколько иное определение односторонней функции, которое утверждает сложность обращения функции на типичном входе.

1.2 Сильные и слабые односторонние функции

Введем следующее обозначение: U_n - равномерное распределение на бинарных строчках длины n.

Определение 1.3. Назовем последовательность положительных чисел ε_n *пренебрежимо малой*, если для любого полинома $p(n)$ для всех достаточно больших n выполняется:

$$\varepsilon_n < \frac{1}{p(n)}.$$

Пример пренебрежимо малой последовательности: $\frac{1}{n^{\log n}}$.

Определение 1.4. Функция $f\colon \{0,1\}^* \to \{0,1\}^*$ называется *(сильно) односторонней* (*one-way function, owf*), если

- f вычислима за полиномиальное время на детерминированной машине Тьюринга,
- для любого многочлена $p(n)$, для любого вероятностного полиномиального по времени алгоритма A, для всех достаточно больших n:

$$\Pr\left[A(f(x), 1^n) \in f^{-1}(f(x))\right] < \frac{1}{p(n)},$$

  где вероятность берётся по равномерно распределенным входным строкам $x \in \{0,1\}^n$ и по случайным числам, используемым алгоритмом A.

Что есть что в данном определении? A - противник, который обращает функцию. Теперь противнику позволяется больше и требуется от него меньше. $f(x)$ - значение функции, которое требуется обратить (то есть нужно найти такой x', что $f(x) = f(x')$). В определении написано "$\in f^{-1}(f(x))$" как раз из-за того, что подходящих значений x может быть несколько. Но противник должен знать какой длины вход ему нужно найти, и мы передаем ему эту длину в палочной системе: 1^n.

Иными словами, функция сильно односторонняя, если вероятность того, что вероятностный полиномиальный алгоритм правильно найдёт прообраз, пренебрежимо мала. В дальнейшем под односторонней функцией будет пониматься сильно односторонняя.

Определение 1.5. Функция $f\colon \{0,1\}^* \to \{0,1\}^*$ называется *слабо односторонней*, если

- f вычислима за полиномиальное время на детерминированной машине Тьюринга,
- *существует* многочлен $p(n)$, такой что для любого вероятностного полиномиального по времени алгоритма A, для всех достаточно больших n:

$$\Pr\left[A(f(x), 1^n) \in f^{-1}(f(x))\right] \leq 1 - \frac{1}{p(n)},$$

  где вероятность опять-таки берётся по равномерно распределенным входным строкам $x \in \{0,1\}^n$ и по случайным числам, используемым алгоритмом A.

Иначе формулу можно переписать следующим образом:

$$\Pr\left[A(f(x), 1^n) \notin f^{-1}(f(x))\right] > \frac{1}{p(n)},$$

Т.е. в случае слабо односторонней функции к функции прилагается конкретная полиномиальная доля, на которой её трудно взломать.

Очевидно, что из существования слабо односторонних функций слежует существование односторонних в наихудшем случае функций. Позже мы докажем, что если существуют слабо односторонние функции, то существуют и сильно односторонние.

1.3 Соглашение о длинах входа

В дальнейшем иногда будет удобно считать, что функция определена не на всех длинах входов, а только на некоторых.

Определение 1.6. Множество I натуральных чисел называется *полиномиально перечислимым*, если функция

$$S_I(1^n) = \min_{i \in I}\{i > n\}$$

полиномиально вычислима.

Теорема 1.2. *Пусть I - полиномиально перечислимое множество, и пусть существует функция f, односторонняя для входов, имеющих длину из множества I (формально в определении 1.4 нужно заменить слова "для достаточно больших n" на для "достаточно больших n из I"). Тогда существует и обычная односторонняя функция.*

Доказательство. Доопределим функцию f на всех остальных входах. А именно, построим ступенчатую функцию $g(x)$, которая:

- совпадает с $f(x)$, если длина x принадлежит множеству I;
- иначе совпадает с $f(x')$, где x' - наибольший префикс x, длина которого лежит в I.

Докажем от противного, что g - обычная односторонняя функция. Пусть A - это взломщик для функции g, и пусть он взламывает эту функцию на бесконечной последовательности длин входов n_k. Построим взломщик для f. Это будет означать, что f - не односторонняя функция, и мы получим противоречие.

Чтобы взломать функцию, достаточно взломать её на бесконечном числе входов. На вход нашему новому взломщику подаётся пара: $(f(x), 1^{i_r})$, где i_r - элемент из множества I. Есть два случая: между i_r и i_{r+1} есть какое-то n_k, на котором мы умеем взламывать функцию g, и такого n_k нет. Будем запускать имеющийся взломщик A (для функции g), подавая ему на вход все возможные длины входов между i_r и i_{r+1} (i_{r+1} можно вычислить за полиномиальное от i_r время пользуясь полиномиальной перечислимостью I):

$$A(f(x), 1^{i_r}), A(f(x), 1^{i_r+1})...A(f(x), 1^{i_{r+1}-1}).$$

Возвращаемое значение x взломщика нам легко проверить на "правильность" (вычислить $f(x)$). Поэтому, если в какой-то момент мы подкинули взломщику A на вход пару $(f(x), 1^{n_k})$, то он взломает её правильно, и мы это обнаружим. Осталось заметить, что так как последовательность $\{n_k\}$ бесконечная, то тех i_r, на которых мы правильно взломаем функцию f, также бесконечное число. (Соответственно, это начала тех промежутков, в которые попадают значения n_k). Таким образом, мы взломали функцию f на бесконечном числе входов, что нам и требовалось. □

Определение 1.7. Функция называется *регулярной*, если она принимает значения одинаковой длины на входах одной длины:

$$\forall x, y \; |x| = |y| \Rightarrow |f(x)| = |f(y)|.$$

Утверждение 1.1. *Если существует односторонняя функция, то существует и регулярная односторонняя функция.*

Доказательство. Пусть g - односторонняя функция. Она вычислима за полиномиальное время, поэтому, существует такой полином $q(n)$, что $|g(x)| \leq q(|x|)$. Тогда регулярной будет следующая функция:

$$f(x) = g(x)10^{q(|x|)-|g(x)|+1}.$$

По "успешному" взломщику для функции f легко построить взломщика для функции g. □

Утверждение 1.2. *Если существует односторонняя функция, то существует и односторонняя функция, сохраняющая длину.*

Доказательство. Продолжаем предыдущую конструкцию. Мы построили функцию:

$$f\colon \{0,1\}^n \to \{0,1\}^{q(n)+1}.$$

Функция h будет функцией, сохраняющей длину:

$$h\colon \{0,1\}^{q(n)+1} \to \{0,1\}^{q(n)+1};$$

$$h(x) = f(y), \text{ где } y = (x)_n \text{ (первые } n \text{ битов строки } x) \text{ , если } |x| = q(n) + 1.$$

□

Упражнение 1.1. В утверждении 1.2 строится функция, определенная только на некоторых длинах входов. Покажите, что можно добиться сохранения длины на всех длинах входов.

1.4 Эквивалентность существования сильной и слабой односторонних функций

Докажем, что сильная и слабая односторонние функции существуют или не существуют одновременно.

Теорема 1.3. *Существует слабая односторонняя функция $\Leftrightarrow$ существует сильная односторонняя фукнция.*

Доказательство. "$\Leftarrow$". Сильная односторонняя функция является слабо односторонней.

"$\Rightarrow$". Построим по слабо односторонней функции f функцию g, которая будет сильно односторонней. Не умаляя общности, можем считать, что f сохраняет длину. Построим g следующим образом:

$$g(x_1, x_2, ...x_m) = (f(x_1), f(x_2), ...f(x_m)),$$

где $m = m(n)$ - некоторый полином от n. На самом деле (как мы увидим позднее) нам достаточно будет взять $m(n) = n^2 \cdot p(n)$, где $p(n)$ - полином из определения слабой односторонней функции f.

Нам нужно доказать, что g - сильно односторонняя. Говоря неформально, при обращении g, нам нужно m раз одновременно (на каждом входе) обратить f. Так как мы в одной точке не всегда это делаем, то m раз одновременно мы совсем с маленькой вероятностью обратим.

Будем доказывать от противного. Пусть существует взломщик для функции g. Это означает, что существует полиномиальный вероятностный алгоритм B и существует многочлен $q(n)$, такие что для бесконечного числа значений n:

$$\Pr\left[B(g(x), 1^{mn}) \in g^{-1}(g(x))\right] \geq \frac{1}{q(n)}.$$

То есть B с маленькой, но полиномиальной вероятностью всё же обращает нашу функцию. (Заметим, что здесь $q(n, m) = q(n)$).

Построим взломщика A' для функции f, который будет её очень часто обращать. Алгоритм A':

В цикле для $1 \leq j \leq m$:

- сгенерировать случайные значения $x_1, x_2, ...x_{j-1}, x_{j+1}, ...x_m$;
- вычислить как результат работы взломщика B:

$$z_1, z_2...z_m = B(f(x_1), ...f(x_{j-1}), y, f(x_{j+1}), ...f(x_m));$$

- если $f(z_j) = y$, то выдать z_j.

Нужно показать, что построенный алгоритм обращает f с хорошей вероятностью. Рассмотрим множество тех входов, на которых алгоритм A' будет работать достаточно хорошо:

$$S_n = \{x \in \{0,1\}^n |\ \Pr\{A'(f(x), 1^n) \in f^{-1}(f(x))\} > \frac{n}{a(n)}\}.$$

Здесь $a(n)$ - некоторый полином, нам будет достаточно: $a(n) = n^2 q(n)$. Мы собираемся повторять наш алгоритм A' много раз, повышая его вероятность успеха. Но если алгоритм на некоторых входах всегда говорит правду, а на некоторых всегда врёт, то повторять его много раз совершенно бессмысленно (мы не увеличим вероятность успеха). Поэтому, мы выделим множество входов, на которых он говорит правду с существенной вероятностью, и покажем, что оно достаточно большое. Скорее всего нам дадут вход из этого множества, а на нем, повторив много раз наш алгоритм, мы добьёмся хорошей вероятности.

Повторив алгоритм A' $a(n)$ раз, получим алгоритм A, претендующий на роль взломщика слабо односторонней функции f. Будем рассматривать вероятность его успеха:

$$\Pr\{A(f(x), 1^n)\} \geq \Pr\{x \in S_n\} \cdot \Pr\{A(f(x), 1^n) |\ x \in S_n\}.$$

Для $x \in S_n$ вероятность "неуспеха" алгоритма A равна вероятности "неуспеха" на каждом из $a(n)$ повторений алгоритма A': $\left(1 - \frac{n}{a(n)}\right)^{a(n)}$. Тогда:

$$\Pr\{A(f(x), 1^n) |\ x \in S_n\} \geq 1 - \left(1 - \frac{n}{a(n)}\right)^{a(n)} = 1 - \Theta(e^{-n}).$$

Предположим, что $\Pr\{x \in S_n\} \geq 1 - \frac{n}{m(n)}$. Вспомним, что $m(n) = n^2 p(n)$. Получаем:

$$\Pr\{A(f(x), 1^n)\} \geq \left(1 - \frac{1}{n\,p(n)}\right)\left(1 - \Theta(e^{-n})\right) > 1 - \frac{1}{p(n)}.$$

Значит, A - это действительно взломщик, и мы получили противоречие с тем, что f - слабо односторонняя.

Теперь рассмотрим случай, когда $\Pr\{x \in S_n\} < 1 - \frac{n}{m(n)}$. Вспомним наше изначальное предположение (что B - это взломщик для функции g) и рассмотрим вероятность его успеха:

$$\Pr\{\text{"успеха } B\text{"}\} = \Pr\{B(g(x_1, ...x_m), 1^{mn}) \in g^{-1}(g(x_1, ...x_m))\} =$$

$$= \Pr\{\text{"успеха } B\text{"} \mid x_1, ...x_m \in S_n\} \cdot \Pr\{x_1, ...x_m \in S_n\} +$$

$$+ \Pr\{\text{"успеха } B\text{"} \mid \exists i\ \ x_i \notin S_n\} \cdot \Pr\{\exists i\ \ x_i \notin S_n\}$$

Вероятность того, что все x_i будут лежать в S_n довольно мала:

$$\Pr\{x_1, ...x_m \in S_n\} < \left(1 - \frac{n}{m(n)}\right)^{m(n)} = \Theta(e^{-n}).$$

Но тогда вероятность успеха B:

$$\Pr\{\text{“успеха } B\text{”}\} \leq \Theta(e^{-n}) + \frac{n}{a(n)} = \Theta(e^{-n}) + \frac{1}{n\,q(n)} < \frac{1}{q(n)}.$$

То есть мы получили противоречие с тем, что B - взломщик! □