2) правила обработки композиции функций;
3) правила работы с локальными переменными и параметрами функции.

ЛИТЕРАТУРА

1. *Стефанцов Д. А., Сафонов В. О., Першин В. В. и др.* Модульный транслятор с языка ЛЯПАС // Прикладная дискретная математика. Приложение. 2016. №8. С. 122–126.
2. `https://github.com/tsu-iscd/lyapas-lcc` — LYaPAS Compiler Chain. 2018.
3. *Агибалов Г. П., Липский В. Б., Панкратова И. А.* О криптографическом расширении и его реализации для русского языка программирования // Прикладная дискретная математика. 2013. №3. С. 93–104.
4. `https://github.com/tsu-iscd/lyapas-lcc/blob/73b21bcd5f674bc6762a379bc32f71f61ee51164/doc/cyaz.md` — LYaPAS Cyaz Documentation. 2018.

УДК 510.52 DOI 10.17223/2226308X/11/41

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ ДИСКРЕТНОГО ЛОГАРИФМА В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ НАД КОНЕЧНЫМИ ПОЛЯМИ[1]

А. Н. Рыбалов

Изучается генерическая сложность проблемы дискретного логарифма в группах точек эллиптических кривых над $\mathrm{GF}(p)$, где p — простое. Доказывается, что её естественная подпроблема генерически трудноразрешима (то есть трудна для почти всех входов) при условии, что проблема дискретного логарифма для эллиптических кривых трудноразрешима в классическом смысле.

Ключевые слова: *генерическая сложность, дискретный логарифм, эллиптическая кривая.*

Введение

Эллиптическая криптография занимается разработкой криптосистем с открытым ключом, основанных на эллиптических кривых над конечными полями. В качестве базиса для этих криптосистем используется проблема дискретного логарифма в группах точек эллиптических кривых над конечными полями. Основное преимущество эллиптической криптографии заключается в том, что на сегодняшний день не известно даже субэкспоненциальных алгоритмов решения проблемы дискретного логарифма на эллиптических кривых, в отличие от проблемы дискретного логарифма в конечных полях. Рассмотрим проблему дискретного логарифма в группах точек эллиптических кривых над конечными полями $\mathrm{GF}(p)$, где p — простое. Эллиптические кривые над такими полями используются в протоколах электронной цифровой подписи ECDSA и ГОСТ Р 34.10-2012.

В [1] развита теория генерической сложности вычислений. В рамках этого подхода алгоритмическая проблема рассматривается не на всём множестве входов, а на некотором подмножестве «почти всех» входов. Такие входы образуют так называемое генерическое множество. Понятие «почти все» формализуется введением естественной меры на множестве входных данных. С точки зрения современной криптографии интересны такие алгоритмические проблемы, которые, являясь (гипотетически) трудными в классическом смысле, остаются трудными и в генерическом смысле, т. е. для почти

[1]Работа поддержана грантом РФФИ, проект №18-41-550001.

всех входов. Это объясняется тем, что при случайной генерации ключей в криптографическом алгоритме происходит генерация входа некоторой трудной алгоритмической проблемы, лежащей в основе алгоритма. Если проблема является генерически легкоразрешимой, то для почти всех таких входов её можно быстро решить и ключи почти всегда будут нестойкими. Поэтому проблема должна быть генерически трудной.

В работе доказывается, что естественная подпроблема проблемы дискретного логарифма в группах точек эллиптических кривых над конечными полями $\mathrm{GF}(p)$ генерически неразрешима за полиномиальное время при условии отсутствия полиномиального вероятностного алгоритма для её решения в худшем случае. Существует правдоподобная гипотеза о том, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, т. е. построить полиномиальный детерминированный алгоритм, решающий ту же задачу. Хотя это пока не доказано, имеются серьезные результаты в пользу этого [2].

1. Генерические алгоритмы

Пусть I есть множество всех входов некоторой алгоритмической проблемы и I_n — множество всех входов размера n. Для подмножества $S \subseteq I$ определим последовательность

$$\rho_n(S) = \frac{|S \cap I_n|}{|I_n|},\ n = 1, 2, 3, \ldots$$

Заметим, что $\rho_n(S)$ — это вероятность попасть в S при случайной и равновероятной генерации входов из I_n. *Асимптотической плотностью* S назовём предел (если он существует)

$$\rho(S) = \lim_{n\to\infty} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$, и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо.

Алгоритм $\mathcal{A}$ с множеством входов I и множеством выходов $J \cup \{?\}$ $(? \notin J)$ называется *генерическим*, если

1) $\mathcal{A}$ останавливается на всех входах из I;
2) множество $\{x \in I : \mathcal{A}(x) =?\}$ пренебрежимо.

Генерический алгоритм $\mathcal{A}$ вычисляет функцию $f : I \to J$, если для всех $x \in I$ $\mathcal{A}(x) = y \in J \Rightarrow f(x) = y$. Ситуация $\mathcal{A}(x) =?$ означает, что $\mathcal{A}$ не может вычислить функцию f на аргументе x. Условие 2 гарантирует, что $\mathcal{A}$ корректно вычисляет f на почти всех входах (входах из генерического множества).

2. Эллиптические кривые и проблема дискретного логарифма

Пусть $p > 3$ — простое число. Эллиптической кривой над конечным полем $\mathrm{GF}(p)$ называется множество точек

$$E = \{(x, y) \in \mathrm{GF}(p)^2 : y^2 = x^3 + ax + b\},$$

где $a, b \in \mathrm{GF}(p)$ такие, что $\Delta = 4a^3 + 27b^2 \neq 0$. К этим точкам добавляется так называемая «точка на бесконечности». На точках эллиптической кривой E вводится операция сложения [3], относительно которой E становится абелевой группой $G(E)$ с нулем — точкой на бесконечности. Для точки $B \in G(E)$ обозначим

$$\langle B\rangle = \{A \in G(E) : A = nB, n \in \mathbb{N}\}.$$

Проблема дискретного логарифма для эллиптических кривых над конечными полями состоит в вычислении функции $dle : I \to \mathbb{N}$, где I — это множество четвёрок (A, B, E, p), таких, что p — простое число, E — фиксированная эллиптическая кривая над $\mathrm{GF}(p)$, B — фиксированная точка из $G(E)$, A — произвольная точка из $\langle B\rangle$. Сама функция dle определяется следующим образом:

$$dle(A, B, E, p) = x \Leftrightarrow xB = A \in G(E).$$

Под размером входа понимается число разрядов в двоичной записи числа p. В настоящее время неизвестно полиномиальных алгоритмов (даже вероятностных) для вычисления функции dle. Это обстоятельство лежит в основе криптостойкости многочисленных криптографических алгоритмов [3].

Для изучения генерической сложности этой проблемы необходимо провести некоторую стратификацию на множестве входов. Рассмотрим любую бесконечную последовательность простых чисел

$$\pi = \{p_1, p_2, \ldots, p_n, \ldots\},$$

удовлетворяющую условию $2^n \leqslant p_n < 2^{n+1}$ для любого n. Будем называть такую последовательность *экспоненциальной*. Теперь определим функцию dle_π как ограничение функции dle на множество четвёрок (A, B, E, p), таких, что $p \in \pi$. Заметим, что для этой функции множество всех входов размера n состоит из четвёрок (A, B, E, p) с фиксированными B, E, p и произвольной точкой $A \in \langle B\rangle$. Очевидно, что проблема вычисления dle_π является подпроблемой вычисления dle. Следующая лемма показывает, что некоторые такие подпроблемы так же трудны, как и оригинальная проблема.

Лемма 1. Если не существует полиномиального вероятностного алгоритма для вычисления dle, то найдётся такая экспоненциальная последовательность простых чисел π, что и для вычисления dle_π нет полиномиального вероятностного алгоритма.

Доказательство. Пусть $P_1, P_2, \ldots$ — все полиномиальные вероятностные алгоритмы. Из предположения о том, что не существует полиномиального вероятностного алгоритма для вычисления dle, следует, что для любого алгоритма P_n существует бесконечно много троек B, E, p, для которых он не может вычислить dle. Из этого следует, что можно выбрать последовательность $\pi' = \{p_1, p_2, \ldots\}$ так, чтобы алгоритм P_n не вычислял dle для B, E, p и для любого n выполнялось бы $p_{n+1} > 2p_n$. Последовательность π' можно расширить до экспоненциальной последовательности π, добавив, где нужно, новые члены. Заметим теперь, что dle_π и будет той функцией, для вычисления которой не существует полиномиального алгоритма. ■

3. Основной результат

Следующий результат говорит о том, что проблема дискретного логарифма для эллиптических кривых над конечными полями остается вычислительно трудной и в генерическом случае при условии её трудноразрешимости в худшем случае.

Теорема 1. Пусть π — любая экспоненциальная последовательность простых чисел. Если существует полиномиальный генерический алгоритм, вычисляющий функцию dle_π, то существует полиномиальный вероятностный алгоритм, вычисляющий dle_π для всех входов.

Доказательство. Пусть существует полиномиальный генерический алгоритм $\mathcal{A}$, вычисляющий функцию dle_π. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$,

вычисляющий dle_π на всём множестве входов. Алгоритм $\mathcal{B}$ на входе (A, B, E, p) работает следующим образом:

1) Сгенерировать случайно и равномерно $y \in \{0, \ldots, p-1\}$ и вычислить $A' = A + yB$.
2) Запустить алгоритм $\mathcal{A}$ на (A', B, E, p).
3) Если $\mathcal{A}(A', B, E, p) = z \in \mathbb{N}$, то $A' = zB = A + yB$, откуда $x = z - y$ — дискретный логарифм для исходной задачи (A, B, E, p).
4) Если $\mathcal{A}(A', B, E, p) =?$, то выдать 0.

Заметим, что алгоритм $\mathcal{B}$ может выдать неправильный ответ только на шаге 4. Докажем, что вероятность этого меньше $1/2$. Действительно, $A' = A + yB$ при $y = 0, \ldots, p-1$ пробегает все элементы $\langle B \rangle$, поэтому множество $\{(A', B, E, p) : y \in \{0, \ldots, p-1\}\}$ совпадает с множеством всех входов размера n. Но алгоритм $\mathcal{A}$ генерический, поэтому доля тех входов (A', B, E, p), на которых он выдаёт неопределённый ответ, стремится к 0 с ростом n и с некоторого момента становится меньше $1/2$. ■

Непосредственным следствием теоремы 1 является следующая

Теорема 2. Если для вычисления функции dle не существует полиномиального вероятностного алгоритма, то существует экспоненциальная последовательность π, такая, что для вычисления функции dle_π не существует генерического полиномиального алгоритма.

ЛИТЕРАТУРА

1. *Kapovich I., Miasnikov A., Schupp P., and Shpilrain V.* Generic-case complexity, decision problems in group theory and random walks // J. Algebra. 2003. V. 264. No. 2. P. 665–694.
2. *Impagliazzo R. and Wigderson A.* P=BPP unless E has subexponential circuits: Derandomizing the XOR lemma // Proc. 29th STOC. El Paso: ACM, 1997. P. 220–229.
3. *Романьков В. А.* Введение в криптографию. 2-е изд., испр. М.: ФОРУМ, 2012. 240 с.

УДК 004.431.4 DOI 10.17223/2226308X/11/42

СОЗДАНИЕ СИСТЕМЫ ТИПОВ ДЛЯ СЕМЕЙСТВА ЯЗЫКОВ АССЕМБЛЕРА

Н. В. Сороковиков

Строится система типов для семейства языков ассемблера, в том числе формально определяются команды, программы и термы языка. Показывается разрешимость задач населённости и проверки типа для ассемблеров с командами `mov` и `jz`.

Ключевые слова: *система типов, ассемблер, статический анализ, бинарные приложения.*

При анализе бинарных приложений исследователи полагаются на различные средства автоматизации. Одной из таких автоматизаций является извлечение типа данных участков памяти процесса. Под типом данных можно понимать стратегию использования участка памяти или, эквивалентно, взаимное расположение разных видов информации относительно друг друга в памяти. Уже существуют и используются средства для определения типов переменных [1], поэтому имеет смысл вопрос, каковы границы применимости статических способов нахождения типов в бинарной программе. Для ответа на этот вопрос можно воспользоваться аппаратом теории типов, а для этого необходимо построить систему типов для языков ассемблера.