Теорема 1. Если для некоммутативной символьной системы уравнений (1) выполнено условие

$$\operatorname{rank}(J(0,0)) = n,$$

то она имеет единственное решение в виде ФСР.

Замечание 1. Из условия теоремы вытекает существование и единственность решения для коммутативного образа системы полиномиальных уравнений; кроме того, оказывается, что из этого условия вытекает также существование и единственность решения исходной некоммутативной символьной системы уравнений (1).

Поскольку ФСР, которые являются компонентами решения системы (1), интерпретируются как формальные языки, порождённые грамматикой (1), то теорема 1 позволяет установить случаи, когда грамматика действительно определяет формальный язык.

ЛИТЕРАТУРА

1. *Egorushkin O. I., Kolbasina I. V., and Safonov K. V.* On solvability of systems of symbolic polynomial equations // Журн. СФУ. Сер. Матем. и физ. 2016. Т. 9. Вып. 2. С. 166–172.
2. *Егорушкин О. И., Колбасина И. В., Сафонов К. В.* Аналог теоремы о неявном отображении для формальных грамматик // Прикладная дискретная математика. Приложение. 2017. № 10. С. 149–151.
3. *Глушков В. М., Цейтлин Г. Е., Ющенко Е. Л.* Алгебра. Языки. Программирование. Киев: Наукова думка, 1973.
4. *Salomaa A. and Soitolla M.* Automata-Theoretic Aspects of Formal Power Series. N.Y.: Springer Verlag, 1978.
5. *Семёнов А. Л.* Алгоритмические проблемы для степенных рядов и контекстно-свободных грамматик // Докл. АН СССР. 1973. № 212. С. 50–52.

УДК 510.52 DOI 10.17223/2226308X/12/56

О ГЕНЕРИЧЕСКОЙ СЛОЖНОСТИ ПРОБЛЕМЫ ДЕКОДИРОВАНИЯ ЛИНЕЙНЫХ КОДОВ[1]

А. Н. Рыбалов

Изучается генерическая сложность проблемы декодирования линейных кодов. Эта проблема лежит в основе известной криптосистемы Мак-Эллиса. Доказывается, что её естественная подпроблема генерически трудноразрешима (то есть трудна для почти всех входов) при условии, что проблема декодирования линейных кодов трудноразрешима в классическом смысле.

Ключевые слова: *генерическая сложность, линейные коды, криптосистема Мак-Эллиса.*

Введение

Криптосистема Мак-Эллиса [1] является одной из первых криптосистем с открытым ключом. В отличие от популярных криптосистем с открытым ключом, использующих теоретико-числовые и алгебраические конструкции [2], система Мак-Эллиса основана на теории кодов, исправляющих ошибки. В этой области были найдены эффективные семейства методов преобразования и восстановления информационных блоков —

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

так называемые коды, которые позволяют исправлять ошибки, возникающие при передаче этих блоков по каналам связи. Наиболее известные коды — это коды Рида — Маллера, Рида — Соломона, Боуза — Чоудхури — Хоквингема, Гоппы и др. Для этих кодов известны эффективные алгоритмы кодирования и декодирования [3]. Идея, лежащая в основе работы криптосистемы Мак-Эллиса, состоит в том, что «хороший» код, для которого известны эффективные алгоритмы декодирования, некоторым образом «маскируется» под более общий код — линейный. Для линейных кодов есть только эффективные алгоритмы кодирования, но неизвестно эффективных алгоритмов декодирования. Более того, доказано [4], что проблема их декодирования является NP-полной, то есть, при условии $\mathrm{P} \neq \mathrm{NP}$, таких эффективных алгоритмов не существует. Криптостойкость системы Мак-Эллиса как раз и основана на трудности проблемы декодирования линейных кодов.

В [5] 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 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 декодирования линейных кодов 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 [6].

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 любого $x \in I$, такого, что $\mathcal{A}(x) \neq?$, имеет место $\mathcal{A}(x) = f(x)$. 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. Проблема декодирования линейных кодов

Пусть G — порождающая $k \times n$-матрица линейного (n, k)-кода над конечным полем $\mathrm{GF}(q)$, исправляющего t ошибок. Проблема декодирования данного линейного кода состоит в вычислении функции $\mathrm{dec}_G : I_G \to \mathrm{GF}(q)^k$, где I_G — это множество векторов $\mathbf{y} \in \mathrm{GF}(q)^n$, таких, что $\mathbf{y} = \mathbf{x}G + \mathbf{e}$, где $\mathbf{x} \in \mathrm{GF}(q)^k$, а $\mathbf{e} \in \mathrm{GF}(q)^n$ такой, что его вес Хэмминга (число ненулевых компонент) $\omega(\mathbf{e}) \leqslant t$. Обозначим также через $I_{G,e}$ множество векторов $\mathbf{y} \in \mathrm{GF}(q)^n$, таких, что $\mathbf{y} = \mathbf{x}G+\mathbf{e}$, где $\mathbf{x} \in \mathrm{GF}(q)^k$. Сама функция dec_G определяется следующим образом:

$$\mathrm{dec}_G(\mathbf{y}) = \mathbf{x}, \text{ если } \mathbf{y} = \mathbf{x}G + \mathbf{e}, \ \mathbf{x} \in \mathrm{GF}(q)^k, \ \omega(\mathbf{e}) \leqslant t.$$

Под размером входа понимается размерность вектора $\mathbf{y}$, то есть кодовая длина n.

Пусть $\mathrm{GF}(q)^* = \bigcup\limits_{k=1}^{\infty} \mathrm{GF}(q)^k$, а I есть множество пар (G, I_G) по всем порождающим матрицам линейных кодов G. Теперь определим проблему декодирования линейных кодов как проблему вычисления функции $\mathrm{dec} : I \to \mathrm{GF}(q)^*$ так, что $\mathrm{dec}(G, I_G) = \mathrm{dec}_G$. В настоящее время неизвестно полиномиальных алгоритмов (даже вероятностных) для вычисления функции dec. Более того, доказано [4], что проблема вычисления этой функции является NP-трудной, то есть, при условии $\mathrm{P} \neq \mathrm{NP}$, таких эффективных алгоритмов не существует.

Для изучения генерической сложности этой проблемы необходимо провести некоторую стратификацию на множестве входов. Рассмотрим любую бесконечную последовательность пар порождающих матриц линейных кодов и векторов ошибок над $\mathrm{GF}(q)$

$$\gamma = \{(G_1, e_1), (G_2, e_2), \ldots, (G_n, e_n), \ldots\},$$

таких, что для любого n имеет место:

1) число столбцов матрицы G_n равно n;
2) размер вектора e_n равен n;
3) $\omega(e_n)$ не превосходит числа ошибок, которые код с матрицей G_n может исправлять.

Определим функцию dec_γ как ограничение функции dec на множество $\bigcup\limits_{(G_n,e_n)\in\gamma} I_{G_n,e_n}$. Очевидно, что проблема вычисления dec_γ является подпроблемой вычисления dec. Следующая лемма показывает, что некоторые такие подпроблемы так же трудны, как и оригинальная проблема.

Лемма 1. Если не существует полиномиального вероятностного алгоритма для вычисления dec, то найдётся такая последовательность пар порождающих матриц линейных кодов и векторов ошибок γ, что и для вычисления dec_γ 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 dec, следует, что для любого алгоритма P_n существует бесконечно много пар (G, e), для которых он не может вычислить dec. Из этого следует, что можно выбрать последовательность $\gamma' = \{(G_1, e_1), (G_2, e_2), \ldots, (G_n, e_n), \ldots\}$ так, чтобы алгоритм P_n не вычислял dec для (G_n, e_n) и чтобы число столбцов матриц в этой последовательности возрастало. В последовательность γ' можно добавить пары так, чтобы для каждого n там была матрица с числом столбцов n — для любого n существует линейный код с повторениями длины n. Получится нужная последовательность γ, такая, что для вычисления функции dec_γ не существует полиномиального алгоритма. ■

3. Основной результат

Следующий результат говорит о том, что проблема декодирования линейных кодов остаётся вычислительно трудной и в генерическом случае при условии её трудноразрешимости в худшем случае.

Теорема 1. Пусть γ — любая последовательность пар порождающих матриц линейных кодов и векторов ошибок. Если существует полиномиальный генерический алгоритм, вычисляющий функцию dec_γ, то существует полиномиальный вероятностный алгоритм, вычисляющий dec_γ для всех входов.

Доказательство. Пусть существует полиномиальный генерический алгоритм $\mathcal{A}$, вычисляющий функцию dec_γ. Построим вероятностный полиномиальный алгоритм $\mathcal{B}$, вычисляющий dec_γ на всём множестве входов. Зафиксируем размер n. Напомним, что множество входов I_{G_n,e_n} размера n есть множество векторов $\mathbf{y} \in \mathrm{GF}(q)^n$, таких, что $\mathbf{y} = \mathbf{x}G + \mathbf{e_n}$, где $\mathbf{x}$ пробегает всё множество $\mathrm{GF}(q)^k$. Алгоритм $\mathcal{B}$ на входе $\mathbf{y}$ работает следующим образом:

1) Генерирует случайно и равномерно $\mathbf{z} \in \mathrm{GF}(q)^k$.
2) Вычисляет $\mathbf{y}' = \mathbf{y} + \mathbf{z}G_n$.
3) Запускает алгоритм $\mathcal{A}$ на $\mathbf{y}'$.
4) Если $\mathcal{A}(\mathbf{y}') =?$, то выдаёт $\mathbf{0}$.
5) Если $\mathcal{A}(\mathbf{y}') = \mathbf{x}'$, то $\mathbf{x}' = \mathbf{x} + \mathbf{z}$. Выдаёт ответ $\mathbf{x} = \mathbf{x}' - \mathbf{z}$ для исходной задачи $\mathbf{y}$.

Заметим, что алгоритм $\mathcal{B}$ может выдать неправильный ответ только на шаге 4. Докажем, что вероятность этого меньше $1/2$. Действительно, множество

$$\{\mathbf{y}' = \mathbf{y} + \mathbf{z}G_n : \mathbf{z} \in \mathrm{GF}(q)^k\}$$

совпадает с множеством I_{G_n,e_n} всех входов размера n. Но алгоритм $\mathcal{A}$ генерический, поэтому доля тех входов $\mathbf{y}'$, на которых он выдаёт неопределённый ответ, стремится к нулю с ростом n и с некоторого момента становится меньше $1/2$. ■

Непосредственным следствием теоремы 1 и леммы 1 является следующее утверждение.

Теорема 2. Если для вычисления функции dec не существует полиномиального вероятностного алгоритма, то существует последовательность пар порождающих матриц линейных кодов и векторов ошибок γ, такая, что для вычисления функции dec_γ не существует генерического полиномиального алгоритма.

ЛИТЕРАТУРА

1. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. 1978. V. 42. No. 44. P. 111–116.
2. *Романьков В. А.* Введение в криптографию. 2-е изд., испр. М.: ФОРУМ, 2012. 240 с.

3. *Рыбалов А. Н.* Введение в теорию кодов, исправляющих ошибки. Омск: Изд-во Ом. ун-та, 2007. 131 с.
4. *Berlekamp E., McEliece R., and Van Tilborg H.* On the inherent intractability of certain coding problems // IEEE Trans. Inform. Theory. 1978. V. 24(3). P. 384–386.
5. *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.
6. *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.