ЛИТЕРАТУРА

1. *Bilgin B., Nikova S., Nikov V., et al.*Threshold implementations of all 3x3 and 4x4 S-boxes // CHES 2012. LNCS. 2012. V. 7428. P. 76–91.

УДК 056.55

АЛГОРИТМ ВОССТАНОВЛЕНИЯ ОТКРЫТОГО ТЕКСТА ПО ШИФРТЕКСТУ В КРИПТОСИСТЕМЕ МАК-ЭЛИСА

А. К. Калужин, И. В. Чижов

Предлагается алгоритм неструктурной атаки на кодовую криптосистему Мак-Элиса с целью дешифрования сообщения, основывающийся на алгоритме Бернштейна — Ланг — Петерса и работающий быстрее любого другого существующего алгоритма неструктурной атаки. Тем самым сделан ещё один шаг в приближении к нижней оценке сложности таких алгоритмов, доказанной М. Финиазом и Н. Сендрие.

Ключевые слова: *криптосистема Мак-Элиса, неструктурные атаки, алгоритм Бернштейна — Ланг —Петерса, алгоритм Шабо — Канто.*

Рассматриваются неструктурые атаки на криптосистему с открытым ключом Мак-Элиса [1] с целью дешифрования сообщения. По сути, решается уравнение $m \cdot G + e = c$, где m и e неизвестны, а $\mathrm{wt}(e) = t$. При этом m — исходное сообщение, G — порождающая матрица кода (открытый ключ), e — вектор ошибки, c — вектор, который подвергается дешифрованию. Найдя вектор ошибки e, мы решим систему полностью, так как вектор m находится из системы линейных уравнений. Все наилучшие алгоритмы неструктурной атаки на систему Мак-Элиса (Штерна, Шабо — Канто и Бернштейна — Ланг — Петерса) основываются на одной идее: итеративно генерируются различные базисы кода и решается задача в предположении, что вектор ошибки e можно выразить через $2p$ (p — параметр алгоритмов) некоторых из зафиксированных векторов базиса.

В 2009 г. М. Финиаз и Н. Синдреир в работе [2] доказали нижнюю теоретическую оценку ожидаемого количества битовых операций, необходимых для дешифрования сообщения в криптосистеме Мак-Элиса. Для кодов Гоппы $(1024, 524, 50)$ (стандартные параметры криптосистемы Мак-Элиса) эта оценка равна $2^{59,9}$. Оценка идеальна и недостижима (в силу предположений при доказательстве). В то же время ожидаемое количество битовых операций, необходимых для дешифрования сообщения, закодированного с помощью этого кода, составляет:

1) для алгоритма Штерна – $2^{66,21}$;
2) для алгоритма Шабо — Канто – $2^{64,1}$;
3) для алгоритма Бернштейна — Ланг — Петерса – $2^{60,55}$.

То есть существующие алгоритмы уже вплотную приблизились к идеальной оценке ожидаемого количества битовых операций.

В работе представляется модификация алгоритма Бернштейна — Ланг — Петерса [3], которая уменьшает как ожидаемое количество итераций, так и ожидаемое количество битовых операций, выполняемых на одной итерации. Достигается это посредством следующих двух оптимизаций.

1) В алгоритме Бернштейна — Ланг — Петерса на каждой итерации фиксируется некоторый базис кода. Он получается из базиса кода, зафиксированного на предыдущей итерации, путём обмена местами c из первых k столбцов матрицы с c столбцами среди оставшихся, с дальнейшим применением модифицированного преобразования

Гаусса. Оптимизация заключается в том, чтобы не менять столбцы, которые менялись на нескольких предыдущих итерациях; тем самым базисы, фиксируемые на итерациях, становятся более независимыми, что немного повышает вероятность найти вектор ошибки e.

2) На очередной итерации алгоритма Бернштейна — Ланг — Петерса ищется вектор ошибки e в виде линейной комбинации ровно $2p$ векторов базиса. Некоторым образом выбираются линейные комбинации из $2p$ векторов базиса, которые образуют векторы-кандидаты. Если вес какого-то вектора-кандидата равен t, то это и есть искомый вектор e. Для проверки необходимо вычислить веса всех векторов-кандидатов. Бернштейн, Ланг и Петерс предлагают считать вес каждого вектора, пока он не превысит t (дальше считать бессмысленно). Оптимизация заключается в том, чтобы отбросить все векторы-кандидаты, у которых среди первых a координат больше чем b координат принимают значение 1. Для остальных осуществить проверку так же, как её делают Бернштейн, Ланг и Петерс. Здесь a и b — новые параметры алгоритма. Смысл оптимизации заключается в следующем: на раннем этапе будет отброшено очень много неподходящих векторов-кандидатов, в то время как подходящий вектор-кандидат может быть отброшен с очень маленькой вероятностью.

Для представленного алгоритма ожидаемое количество битовых операций, необходимых для дешифрования сообщения, закодированного с помощью кодов Гоппы $(1024, 524, 50)$, равно $2^{60,1}$. Это на 27,5 % меньше, чем для алгоритма Бернштейна — Ланг — Петерса, самого быстрого из существующих алгоритмов неструктурной атаки на криптосистему Мак-Элиса. Тем самым осуществлено ещё большее приближение к теоретической оценке количества битовых операций при дешифровании сообщения.

ЛИТЕРАТУРА

1. *McEliece R. J.* A public-key cryptosystem based on algebraic coding theory // DSN Progress Report. January and February 1978. No. 42–44. P. 114–116.
2. *Finiasz M. and Sendrier N.* Security bounds for the design of code-based cryptosystems // Asiacrypt'2009. LNCS. 2009. V. 5912. P. 88–105.
3. *Bernstein D. J., Lange T., and Peters C.* Attacking and defending the McEliece cryptosystem // Post-Quantum Cryptography: Second International Workshop, PQCrypto 2008. Cincinnaty, OH, USA. October 17–19, 2008. P. 31–46.

УДК 519.17, 004.056.2, 004.056.53

О ВЕРОЯТНОСТНЫХ ХАРАКТЕРИСТИКАХ СЛУЧАЙНЫХ ГРАФОВ, ПОРОЖДАЕМЫХ АЛГОРИТМАМИ ПОИСКА КОЛЛИЗИЙ КРИПТОГРАФИЧЕСКИХ ХЭШ-ФУНКЦИЙ

Г. А. Карпунин

Описывается теоретико-графовая модель некоторых алгоритмов поиска коллизий хэш-функций SHA-1 и RIPEMD, и в данной модели выводится точная формула средней трудоёмкости этих алгоритмов.

Ключевые слова: *криптографические хэш-функции, коллизии, случайные графы.*

В алгоритмах поиска коллизий некоторых хэш-функций семейства MDx (см., например, SHA-1 [1] и RIPEMD [2, 3]), встречается процедура $\mathcal{A}$, которую можно смоделировать случайным процессом Γ_N. Данный случайный процесс строит корневое