

УДК 519.7

DOI 10.17223/2226308X/13/40

КОМПАКТНЫЙ ТРАНСЛЯТОР АЛГОРИТМОВ В БУЛЕВЫ ФОРМУЛЫ ДЛЯ ПРИМЕНЕНИЯ В КРИПТОАНАЛИЗЕ¹

Д. А. Софронова, К. В. Калгин

Представлен транслятор, позволяющий преобразовывать описание криптографической задачи (криптоанализ шифра или хэш-функции, поиск APN-функций) в КНФ. В дальнейшем SAT-решатель устанавливает истинность формулы и находит набор, выполняющий КНФ. Отличительные особенности данной разработки — универсальность, малый объём исходного кода (300 строк C++), легко модифицируемая и расширяемая реализация.

Ключевые слова: криптоанализ, SAT-решатель, атака «угадай-и-вычисли».

В основе одного из методов анализа симметричных шифров лежит использование SAT-решателей. По алгоритму, задающему криптографическую функцию, строится КНФ. Если для данной КНФ удастся найти выполняющий набор, то имеем решение задачи криптоанализа. SAT-задача — задача определения выполнимости логической формулы [1]. SAT-решатель — программа, которая ищет набор значений переменных, на котором формула истинна. Известно, что эта задача NP-полная. Несмотря на это, для множества практических задач SAT-решатели определяют выполнимость формул с тысячами переменных за приемлемое время. Для проведения криптоанализа с помощью SAT-решателя необходим механизм представления криптографических алгоритмов в виде КНФ в формате DIMACS.

На данный момент существует несколько разработок, позволяющих на выходе получать КНФ. Приведём краткое описание двух разработок, специализирующихся на криптоанализе шифров — Grain of Salt [2] и Transalg [3].

Transalg универсален и позволяет сводить к задаче выполнимости не только криптографические задачи, но и некоторые задачи биоинформатики. Шифр описывается на специальном си-подобном языке, после чего строится КНФ. В настоящее время при помощи Transalg получены SAT-кодировки многих симметричных алгоритмов, а также хэш-функций [4]. Являясь полноценным транслятором, Transalg анализирует текст описания с помощью лексического, синтаксического и семантического анализаторов, что делает его достаточно сложным для модификации и расширения.

Grain of Salt (GoS) — программный комплекс описания поточных шифров на базе регистров сдвига и последующего автоматического проведения атаки «угадай-и-вычисли», который разработал автор cryptominisat [5] M. Soos. Данный вариант хорошо оптимизирован с помощью карт Карно, поэтому выходная КНФ имеет меньший размер по сравнению с КНФ, полученной без оптимизации. Построены SAT-кодировки шифров Grain, Trivium, Bivium, Crypto1 и Hitag2 [2].

В данной работе представлен программный комплекс, универсальный, легко расширяемый, простой и понятный для пользователей (в том числе на уровне реализации). Под криптографическими задачами далее подразумеваем не только задачи анализа шифров и хэш-функций, но и задачи поиска APN-функций, определения EA-эквивалентности булевых и векторных функций.

¹Работа выполнена при поддержке Математического центра в Академгородке (г. Новосибирск), соглашение с Министерством науки и высшего образования Российской Федерации № 075-15-2019-1613, и Лаборатории криптографии JetBrains Research.

Основная идея заключается в том, что криптографическая задача (алгоритм или множество ограничений) описывается на языке C++ с использованием специальных классов `varBool` и `varInt`, у которых переопределены все операторы. Полиморфизм в C++ позволяет переопределить работу операторов для новых типов так, что при выполнении некоторых действий над данными происходит формирование КНФ (в зависимости от операций добавляются разные конструкции) или реальное исполнение алгоритма. Через указание параметров при компиляции можно получить реализацию исходного алгоритма или генератор, получающий на выходе КНФ. Кроме того, есть возможность задать значения определённых переменных `varBool` до генерации КНФ, например для частичного задания битов ключа при проведении атаки «угадай-и-вычисли». При использовании C-интерфейса SAT-решателя `cryptominisat` можно запускать решатель без промежуточной записи КНФ в файл. Работа программы построена на операциях, обрабатывающих новые типы и неявно формирующих КНФ на основе логики операций. Немаловажным плюсом является то, что большинство шифров описаны на языке C. Построение задачи криптоанализа таких шифров легко осуществляется в проекте заменой типов данных в коде. Аналогичным образом преобразуются алгоритмы, описанные на языке `TransAlg`. Программа является гибкой, использование возможностей языка C++ (циклы, условные операторы, классы, шаблоны) позволяет описывать алгоритмы разной сложности. На данном этапе с использованием транслятора и описанных в нём механизмов регистров сдвига построены SAT-кодировки шифров, описанных в [2], а также генератор A5/1.

ЛИТЕРАТУРА

1. *Otpuschennikov I., Semenov A., Gribanova I., et al.* Encoding cryptographic functions to SAT using TRANSALG system // Proc. ECAI'16. IOS Press, 2016. P. 1594–1595.
2. *Biere A., Heule M., Maaren H., and Walsh T.* Handbook of Satisfiability. IOS Press, 2009. 966 p.
3. *Semenov A., Otpuschennikov I., Gribanova I., et al.* Translation of algorithmic descriptions of discrete functions to SAT with application to cryptanalysis problems // Log. Methods Comput. Sci. 2020. V. 16. Iss. 1. P. 29:1–29:42.
4. *Soos M., Nohl K., and Castelluccia C.* Extending SAT solvers to cryptographic problems // LNCS. 2009. V. 5584. P. 244–257.
5. *Soos M.* Grain of salt — an automated way to test stream ciphers through SAT solvers // Tools. 2010. V. 10. P. 131–144.