Раздел II. Безопасность информационных технологий

УДК 681.03.245

Л.К. Бабенко, Е.А. Ищукова, Е.А. Маро, И.Д. Сидоров, П.П. Кравченко

РАЗВИТИЕ КРИПТОГРАФИЧЕСКИХ МЕТОДОВ И СРЕДСТВ ЗАЩИТЫ ИНФОРМАЦИИ

Рассмотрены основные моменты построения и анализа современных криптографических систем. Для симметричных алгоритмов шифрования широко рассмотрена возможность применения методов дифференциального и алгебраического анализов. Для асимметричных систем рассмотрены алгоритмы анализа на основе методов факторизации и дискретного логарифмирования. Также рассмотрены подходы к анализу современных функций хэширования. В работе приведены основные сведения, полученные коллективом кафедры БИТ, по каждому из направлений исследования, в том числе и с использованием распределенных многопроцессорных вычислений.

Криптография; криптоанализ; симметричное шифрование; дифференциальный криптоанализ; алгебраический анализ; асимметричное шифрование; факторизация; дискретное логарифмирование; функции хэширования.

L.K. Babenko, E.A. Ishchukova, E.A. Maro, I.D. Sidorov, P.P. Kravchenko

DEVELOPMENT OF CRYPTOGRAPHIC TECHNIQUES AND MEANS OF INFORMATION SECURITY

The paper discusses the main points of construction and analysis of modern cryptographic systems. For symmetric encryption algorithms are widely considered the possibility of applying the methods of differential and algebraic analysis. For asymmetric systems considered analysis algorithms based on factorization and discrete logarithms. Also, the approaches to the analysis of modern hashing functions are considered. The article consist of investigation for each sphere of information security, which were gotten by staff of the Department of Information security . Also authors have observed applying of distributed multiprocessor computing on information security systems.

Cryptography; cryptanalysis; symmetric encryption; differential cryptanalysis; algebraic analysis; asymmetric encryption; factorization; discrete logarithm; hash functions.

Стремление защитить свои интересы было присуще человеку с давних пор. Еще в древности он использовал различные варианты кодирования информации, изобретал устройства, которые бы способствовали сохранению в тайне секретной информации и при этом обеспечивали легкость ее шифрования - расшифрования.

Можно сказать, что толчком для развития теории информации в ее современном понимании стала работа Огюста Кергоффса «Военная криптография», опубликованная в 1883 г. Позднее Клод Шеннон в своей работе «Теория связи в секретных системах», опубликованной в 1949 г., сформулировал основные постулаты теоретической криптографии. Именно он определил, какими свойствами должны обладать надежные шифры, ввел в криптографию понятия перемешивания и рассеивания и предложил формировать криптографически стойкие системы на основе простых математических преобразований.

Долгое время криптография оставалась секретной наукой, в тайны которой был посвящен лишь узкий круг лиц. Это было естественно, так как в первую очередь она была направлена на сохранение государственных секретов. Ситуация стала меняться во второй половине XX века с появлением персональных компьютеров. Когда практически каждый человек получил возможность оперировать электронной информацией, возникла естественная потребность как-то защищать эту информацию от несанкционированного доступа. Широкое распространение получило использование симметричной криптографии, а несколько позднее и асимметричной. Также важную роль в современной криптографии играют поточные шифры и функции хэширования.

Современная криптография основана на понятии односторонней функции $y= f_k(x)$, которая обладает следующим важным свойством: Зная x и k, легко вычислить значение y, но при этом вычислительно сложно определить значение x, зная только y. Стойкость современных шифров, помимо собственно алгоритма шифрования, определяется длиной используемого ключа шифрования. Современная криптография исходит из того, что секретность шифра обеспечивается исключительно ключом шифрования, так как сам алгоритм рано или поздно может стать известным противнику.

В настоящей работе мы постараемся осветить наиболее характерные особенности современных криптографических систем, а также рассмотрим основные проблемы, связанные с определением криптографической стойкости современных систем защиты информации, и подходы к их решению.

Симметричная криптография. Симметричное шифрование, которое в литературе еще называют традиционным шифрованием или шифрованием с общим ключом, до изобретения шифрования с открытым ключом было единственным методом шифрования. В 1976 г. в США был утвержден стандарт шифрования данных DES (Data Encryption Standard). Этот стандарт использовался довольно длительное время (более 20 лет), пока в 2001 г. не был принят новый стандарт AES (Advanced Encryption Standard). В основу последнего лег алгоритм шифрования Rijndael. В России же официальным государственным стандартом является алгоритм шифрования ГОСТ 28147-89.

Для симметричных алгоритмов шифрования характерны следующие свойства:

- использование одного и того же алгоритма как для зашифрования, так и для расшифрования данных;
- использование одного ключа, который хранится в секрете.

Современные симметричные алгоритмы шифрования разделяются на блочные и поточные. Для блочных алгоритмов шифрование информации производится небольшими порциями – блоками; как правило, размер блока кратен 32 битам и составляет 64, 128, 192 или 256 битов. К современным алгоритмам симметричного шифрования относятся такие шифры, как DES, AES (Rijndael), RC5, ГОСТ 28147-89, и многие другие. Подробное описание многих современных симметричных алгоритмов шифрования можно найти в монографии авторов настоящей статьи «Современные алгоритмы шифрования и методы их анализа» [1]. Кроме того, в 2009 г. вышла книга С. Панасенко «Алгоритмы шифрования», представляющая собой довольно подробный справочник по симметричным системам шифрования, в котором описано более 50 алгоритмов шифрования.

Поточные шифры обычно шифруют информацию в режиме реального времени, как правило, побитно (реже побайтно) и используют для шифрования специально вырабатываемую псевдослучайную последовательность. К поточным шифрам, например, относится широко известный шифр A5/1, который используется для шифрования связи GSM. В настоящее время имеется достаточно большое число различных поточных шифров. Только в книге «Поточные шифры», вышедшей в 2003 г., описано более 20 шифров.

В настоящий момент выделяют два основных способа построения симметричных алгоритмов шифрования: схему Фейстеля и сеть на основе подстановок и перестановок (SPN – Substitution-Permutation Network). По схеме Фейстеля построены алгоритмы DES, RC5, ГОСТ 28147-89 и др. Самым ярким представителем использования сети SPN является стандарт AES.

Ключевой задачей защиты информации является создание стойких алгоритмов шифрования. Любой конструируемый алгоритм подвергается тщательному анализу с целью выявления его слабых мест и возможности взлома. Алгоритм является относительно стойким до тех пор, пока не будут обнаружены методы и пути его анализа, позволяющие получить секретный ключ шифрования значительно быстрее, чем это можно сделать с использованием метода «грубой силы» или «полного перебора». Рассмотрим основные известные на сегодняшний день методы анализа симметричных систем.

Дифференциальный криптоанализ. Метод дифференциального криптоанализа (ДК) впервые был предложен в начале 90-х годов прошлого века Э. Бихамом и А. Шамиром для анализа алгоритма шифрования DES. Хотя в книге Б. Шнайера [2] упоминается о том, что разработчики алгоритма DES знали о возможности такого анализа еще во время разработки алгоритма в 70-х годах XX века, широкая общественность узнала о дифференциальном криптоанализе именно из работы [3]. Метод ДК оказался первым методом, позволяющим взломать DES при оценке сложности задач менее 2^{55}. Согласно [3], с помощью данного метода можно провести криптоанализ DES при усилиях порядка 2^{37}, но при наличии 2^{47} вариантов избранного открытого текста. Хотя 2^{37}, очевидно, значительно меньше, чем 2^{55}, необходимость при этом иметь 2^{47} вариантов избранного открытого текста превращает данный вариант схемы криптоанализа в чисто теоретическое упражнение. Это связано с тем, что метод ДК был известен в момент разработки DES, но засекречен по очевидным соображениям, что подтверждается публичными заявлениями самих разработчиков [2]. Было показано, что если поменять порядок следования блоков замены в алгоритме шифрования DES или использовать другие наборы таблиц подстановок и перестановок, то алгоритм становится сразу намного слабее и может быть взломан менее чем за половину времени, требуемого для анализа алгоритма DES с помощью полного перебора.

С помощью метода дифференциального криптоанализа (differential cryptanalysis), предложенного Э. Бихамом и А. Шамиром [3], сложность анализа сократилась до 2^{37}. Однако при этом для проведения анализа необходимо было иметь 2^{37} особым образом подобранных текстов, зашифрованных на одном и том же секретном ключе. Дальнейшее развитие этого метода показало возможность его применения к целому классу различных видов шифров, позволило выявить слабые места многих используемых и разрабатываемых алгоритмов шифрования. Сегодня этот метод, а также некоторые его производные, такие как метод линейно-дифференциальный, метод невозможных дифференциалов, метод бумеранга, широко используются для оценки стойкости вновь создаваемых шифров.

Само название “дифференциальный криптоанализ” происходит от английского слова difference, т.е. разность. Именно поэтому в отечественной литературе этот вид анализа еще иногда называют разностным методом. Исходя из названия, можно понять, что при рассмотрении возможности анализа некоторого блочного алгоритма шифрования ученым пришло в голову использовать не отдельные тексты, а пары текстов. Понятно, что два текста будут иметь различия в некоторых позициях. Для того чтобы определить это различие, достаточно пару текстов сложить между собой по модулю два. Результат такого сложения даст на выходе значение 0 в тех позициях, в которых исходные тексты были равны между собой, и

соответственно значение 1 в тех позициях, в которых исходные тексты отличались. Например, рассмотрим два 4-битовых сообщения: X = 0011 и X' = 1010. В результате сложения текстов X и X' была получена разность ΔX = 1001, полученное значение ΔX принято называть дифференциалом или разностью. В дифференциальном криптоанализе значение разности (дифференциала) принято обозначать символом Δ. Разность, полученная в результате сложения текстов X и X', показывает, что во второй и третьей позициях исходные сообщения X и X' были равны, а в первой и четвертой отличались друг от друга.

В общем виде дифференциальный анализ блочных алгоритмов шифрования сводится к следующим основным этапам. Первый этап: нахождение для алгоритма шифрования характеристик, обладающих максимальными значениями. Поиск характеристик ведется на основе дифференциальных свойств нелинейных криптографических примитивов, входящих в состав алгоритма шифрования. Второй этап: поиск правильных пар текстов с использованием найденных характеристик. Третий этап: анализ правильных пар текстов и накопление статистики о возможных значениях секретного ключа шифрования.

Первый пункт, заключающийся в поиске лучших характеристик для большинства алгоритмов, выполняется единожды и является теоретической задачей. Значения характеристик полностью зависят от структуры алгоритма шифрования и используемых криптографических примитивов. Иначе дело обстоит лишь с теми алгоритмами, которые обладают нефиксированными элементами. К таким алгоритмам можно, например, отнести алгоритм шифрования ГОСТ 28147-89, у которого S-блоки замены могут выбираться произвольным образом. Для таких алгоритмов поиск характеристик необходимо каждый раз начинать сначала, основываясь на дифференциальных свойствах выбранных S-блоков. Для автоматизации процесса анализа можно разработать алгоритм поиска лучших характеристик, основываясь на алгоритмах поиска по дереву [5]. Для таких алгоритмов можно использовать параллельные модели для ускорения поиска характеристик.

Второй шаг анализа является вычислительно стойкой задачей для любого алгоритма шифрования, при этом не важно, обладает он фиксированными или нефиксированными элементами. Анализ заключается в опробовании большого числа пар текстов с целью определения правильной пары текстов, т.е. той пары текстов, которую в дальнейшем можно использовать для анализа с целью поиска секретного ключа шифрования. Данный шаг может быть легко представлен в виде параллельных вычислений для сокращения времени анализа [4, 5].

Последний шаг легко реализуем, требует гораздо меньше вычислений в сравнении со вторым шагом. Он может быть реализован как отдельно в виде последовательного алгоритма, так и быть включенным в состав параллельных алгоритмов по поиску правильных пар текстов. В последнем случае при нахождении правильной пары текстов сразу можно провести ее анализ по накоплению статистики о возможном значении секретного ключа.

На кафедре БИТ ТТИ ЮФУ исследования в области дифференциального криптоанализа ведутся с 2003 г. За это время получено множество результатов, которые нашли отражение в большом числе публикаций.

На основе метода дифференциального криптоанализа, предложенного Э. Бихамом и А. Шамиром, разработаны последовательные алгоритмы поиска правильных пар текстов по заданному дифференциалу и секретного ключа для проведения анализа n-раундового ($n \leq 16$) алгоритма DES. Проведен анализ шести раундов алгоритма DES с использованием наиболее вероятных значений дифференциалов [7]. Показано, что на 2-процессорной системе с частотой процессоров 1,41 ГГц время анализа в среднем составляет 7,5 минут, на 16-процессорной – 56 секунд.

Проведен полный анализ диапазона входных разностей для алгоритма DES, состоящего из 8, 10, 12, 14 и 16-ти раундов на m-процессорном кластере ($m \leq 16$) с использованием разработанной методики. Показано, что при увеличении числа процессоров наблюдается практически линейный рост ускорения времени анализа. Кроме того, показано, что процент текстов, полностью соответствующих схеме преобразования заданного дифференциала, от общего числа найденных правильных пар текстов, лежит в диапазоне от 80 до100 %, что гарантирует успех анализа. Проведен анализ 16-раундового алгоритма DES с использованием 16-процессорного кластера (частота 1,41 ГГц). Время работы программы составило 24 часа 13 минут.

Разработан рекурсивный алгоритм поиска дифференциалов, обладающих максимальной вероятностью, для алгоритма шифрования ГОСТ 28147-89, учитывающего различные варианты заполнения для блоков замены, для отбора правильных пар текстов при дальнейшем анализе [5]. На его основе разработан параллельный алгоритм поиска наиболее вероятных дифференциалов для алгоритма ГОСТ 28147-89 с учетом статического и динамического распределения данных и межпроцессорных взаимодействий при выявлении дифференциала с максимальной вероятностью. Проведены тестовые испытания анализа алгоритма шифрования ГОСТ 28147-89 для различных сочетаний следующих параметров: числа раундов шифрования, начального значения пороговой вероятности, количества процессоров, способа распределения данных. Показано, что при задании ненулевого значения пороговой вероятности, скорость вычислений в среднем возрастает в 1,285 раз. Показано, что при использовании 16-ти процессоров для анализа со статическим распределением данных время вычислений сокращается в 2,88 раза, а с динамическим – в 4,4 раза по сравнению с такими же расчетами на двухпроцессорной системе. Показано, что для алгоритма с динамическим распределением данных до 8-ми процессоров наблюдается линейный рост ускорения.

Для алгоритма ГОСТ 28147-89 показано, что существует ряд S-блоков, обладающих слабыми свойствами по отношению к дифференциальному криптоанализу [6]. Использование таких блоков в алгоритме ГОСТ позволяет получать характеристики, обладающие довольно высокими вероятностями, которые можно использовать для проведения атаки. Так, при использовании одного и того же слабого блока замены, вероятность характеристики для 32-х раундов ГОСТ может составлять 2^{-25}, что позволяет сравнительно легко получать правильные пары текстов для анализа. Для подтверждения предположений была осуществлена атака на 12 раундов алгоритма ГОСТ, которая за несколько минут позволяет определить первый раундовый подключ шифрования.

Разработан параллельный алгоритм проведения дифференциального анализа алгоритма Rijndael, лежащего в основе стандарта шифрования данных AES, с учетом межпроцессорного распределения данных и взаимодействия процессов при нахождении ключа шифрования. Рассмотрена возможность применения метода дифференциального криптоанализа к анализу поточных шифров [7] и современных функций хэширования [8]. Дополнительные сведения о дифференциальном криптоанализе можно найти в монографиях [1, 9].

Алгебраический анализ. Сущность алгебраических методов анализа заключается в получении уравнений, описывающих нелинейные преобразования замены S-блоков, с последующим решением найденных систем уравнений и получением ключа шифрования. Данный метод криптоанализа относится к атакам с известным открытым текстом, для успешного анализа достаточно иметь одну пару открытый текст/шифртекст. Алгебраические методы криптоанализа состоят из следующих этапов:

- составление системы уравнений, описывающей преобразования в нелинейных криптографических примитивах анализируемого шифра (чаще всего для симметричных алгоритмов шифрования такими нелинейными компонентами являются S-блоки замены);
- решение полученной системы уравнений.

Рассмотрим подробнее первый этап алгебраического криптоанализа. Для шифров, подобных Rijndael, при составлении уравнений используется таблица замены S-блоков. Ограничимся рассмотрением одночленов, состоящих из произведения двух переменных. Тогда уравнения, описывающие работу S-блоков, имеют вид

$$\sum \alpha_{ij} x_i x_j + \sum \beta_{ij} y_i y_j + \sum \gamma_{ij} x_i y_j + \sum \delta_i x_i + \sum \varepsilon_i y_i + \eta = 0,$$

где $x_i x_j$ – комбинация входных битов S-блока;

$y_i y_j$ – комбинация выходных битов S-блока;

$x_i y_j$ – комбинация входных и выходных битов;

x_i и y_i – соответственно входные и выходные биты S-блока;

η – коэффициент, принимающий значения 0 или 1.

При получении уравнений нужно рассмотреть все возможные комбинации данных одночленов. В случае, когда число бит на входе S-блоков равно s, получаем, что число одночленов, встречающихся в системе, вычисляется по формуле $t = \binom{2s}{2} + 2s + 1$ и включает в себя входные и выходные значения S-блока (2s), все их возможные произведения $\binom{2s}{2}$ и коэффициент η. Число всех возможных комбинаций одночленов составляет 2^t. Для произвольного блока замены число линейно независимых уравнений $r \geq t - 2^s$.

Для проверки всех полученных комбинаций на соответствие заданному S-блоку требуется составить таблицу истинности на основании замен, выполняемых в исследуемом S-блоке. Для проверки комбинаций на соответствие таблице истинности следует осуществить построковую подстановку значений одночленов из таблицы и выполнить операцию сложения по модулю 2. Таким образом, для каждой комбинации выполняется подстановка и сложение для всех возможных входных значений S-блока (2^s раз). Результаты суммирования сравниваются с нулем. Если для всех строк таблицы истинности равенство оказывается верным, то уравнение, заданное данной комбинацией одночленов, удовлетворяет таблице замены исследуемого S-блока, и его следует отобрать для составления искомой системы. Далее необходимо провести анализ уравнений и выбрать для формирования системы линейно независимые уравнения, содержащие минимальное число нелинейных элементов.

Второй этап алгебраического криптоанализа заключается в решении системы. В криптоанализе разработаны различные подходы к решению нелинейных систем булевых уравнений. Наиболее эффективными, как показывает практика криптоанализа, являются методы, использующие линеаризацию исходной системы.

XL-метод (eXtended Linearization) предложили Nicolas Courtois, Alexander Klimov, Jacques Patarin и Adi Shamir в работе [10]. Пусть имеется нелинейная система, содержащая m уравнений и 2s переменных. XL-метод базируется на умножении каждого уравнения 1…m на произведения переменных степени, меньшей или равной D–2. Рассмотрим вычисление параметра D алгоритма XL атаки. При умножении исходных уравнений системы на одночлены степени $\leq$(D–2) получаем

примерно $R \approx \binom{2s}{\mathrm{D}-2} m$ новых уравнений. Общее число одночленов, встречающихся в этих уравнениях, составляет $T = \binom{2s}{D}$. Так как система будет решаться способом линеаризации, т. е. путем замены всех нелинейных одночленов на новые переменные, необходимо чтобы число уравнений было больше числа одночленов $R = \binom{2s}{\mathrm{D}-2} m \geq \binom{2s}{D} = T$. Отсюда получаем, что $m \geq \binom{2s}{D} \Big/ \binom{2s}{\mathrm{D}-2} / \approx (2s)^2 / D^2$. Следовательно, $D \approx \frac{2s}{\sqrt{m}}$. При этом должно выполняться условие D>2, иначе не будет получено новых уравнений, так как степень отобранных для умножения уравнений одночленов, определяемая разностью D–2, будет равна нулю.

Алгоритм XL-метода состоит из двух шагов:

- Multiply: умножение каждого уравнения исходной системы на произведение переменных в степени ≤ D–2.
- Linearize: замена каждого одночлена в степени ≤ D на новую переменную и применение метода исключения Гаусса.

Сложность анализа заключается в построении системы всех возможных линейных уравнений и последующего ее решения. Для ускорения процесса анализа построение уравнений для системы можно производить параллельно. Также переопределенную систему многих уравнений целесообразно решать с использованием параллельных вычислений с последующим объединением результата.

Для проведения исследований в области алгебраического криптоанализа на кафедре БИТ ТТИ ЮФУ в качестве объекта исследования был выбран алгоритм шифрования ГОСТ 28147-89. Наибольшую сложность в его анализе представляет использование операции целочисленного сложения по модулю 2^n. В связи с этим для начала была рассмотрена упрощенная схема для алгоритма ГОСТ, в которой операция целочисленного сложения по модулю 2^n была заменена на операцию сложения по модулю 2 (алгоритм ГОСТ⊕). На основе полученных данных был предложен алгоритм анализа оригинальной версии алгоритма ГОСТ [11, 12].

В ходе исследования показано, что для 32-х раундов алгоритма шифрования ГОСТ составлена система из 5376 квадратных уравнений, связывающих входы и выходы блоков замены. Общее число переменных равно 2048, в системе содержится 9472 одночлена. Процесс генерации системы уравнений для одного раунда ГОСТ потребует около 12 часов (исследование проводилось с использованием процессора AMD Athlon 64 X2DualCoreProcessor 3800+, RAM 1G).

Асимметричная криптография. Традиционно считается, что концепция асимметричной криптографии впервые была предложена в 1976 г. Уитвелдом Диффи (Whitfield Diffie) и Мартином Хеллманом (Martin Hellman) и опубликована в том же году в основополагающей работе "Новые направления в криптографии" (“New Directions in Cryptography”). К числу отцов-основателей асимметричной криптографии относят также и Ральфа Меркля (Ralph Merkle), который независимо от Диффи и Хеллмана пришел к тем же конструкциям, однако опубликовал свои результаты только в 1978 г.

С 1976 г. было создано множество криптографических алгоритмов, использующих концепцию открытых ключей. Многие из них не являются стойкими, а многие стойкие алгоритмы очень часто не пригодны для практической реализации, поскольку в них используется слишком большой ключ либо размер полученного с их помощью шифртекста намного превышает объем открытого текста. И только

весьма небольшая часть указанных алгоритмов являются и стойкими, и пригодными для практического использования. Как правило, эти алгоритмы основываются на решении одной из трудных математических задач, таких как задача дискретного логарифмирования или задача факторизации больших чисел. Известны всего лишь три алгоритма, которые предоставляют достаточные возможности как для шифрования текста, так и для его цифровой подписи: RSA, Эль-Гамаля и Рабина. Однако все эти алгоритмы работают достаточно медленно, зашифровывая и расшифровывая данные значительно медленнее, чем симметричные алгоритмы. В результате они часто непригодны для шифрования больших объемов данных, а используются для пересылки короткой шифрованной информации. Например, секретного ключа шифрования для симметричных криптосистем.

Для асимметричных алгоритмов шифрования характерны следующие свойства:

- не обязательно использование одного и того же алгоритма как для зашифрования, так и для дешифрования данных;
- использование двух ключей, один из которых является открытым, а второй – секретным.

Для анализа асимметричных криптосистем на сегодняшний день существует достаточно большое разнообразие методов. Среди них наиболее известны такие, как метод Гельфонда, «giant step-baby step», метод встречи на случайном дереве, метод базы разложения, метод решета числового поля, метод Ферма, метод непрерывных дробей, метод квадратичного решета и другие. Однако если при анализе симметричных криптосистем различные методы используют различные приемы, такие как линеаризация, разностные характеристики пар текстов, составление систем переопределенных уравнений, то при анализе асимметричных криптосистем все методы сводятся к решению двух задач различными способами – задачи дискретного логарифмирования и задачи факторизации больших чисел.

Задача дискретного логарифмирования в группе F_q^* формулируется следующим образом. Пусть F_q – конечное поле из $q = p^n$ элементов. Для образующей a подгруппы простого порядка r группы F_q^* и экспоненты b необходимо найти показатель x такой, что $a^x = b$. Эта задача может решаться как универсальными методами логарифмирования («giant step – baby step», метод Полларда) в произвольной конечной циклической группе вычислимого порядка, так и специальными методами для группы F_q^*.

Задача факторизации, или другими словами, задача разложения составного числа на множители, является одной из первых задач, использованных для построения криптосистем с открытым ключом. Эта задача формулируется так: для данного положительного целого числа n найти его каноническое разложение $n = p_1^{\alpha_1} p_2^{\alpha_2} \ldots p_s^{\alpha_s}$, где p_i – попарно различные простые числа, $\alpha_i \geq 1$.

Обе эти задачи широко исследуются научным коллективом кафедры БИТ ТТИ ЮФУ [13, 14]. Для исследования дискретного логарифмирования были выбраны следующие методы.

1. Для работы в мультипликативной группе числового поля – методы базы разложения и решета числового поля.

2. Для работы в группе точек эллиптической кривой – методы встречи посередине и встречи на случайном дереве.

Эти методы выбраны из-за хороших асимптотических оценок и возможности эффективного распараллеливания. Для каждого из методов разработаны алгоритмы, позволяющие распараллелить определённые этапы вычислений. Для методов, работающих в мультипликативной группе числового поля, разработаны алгоритмы распараллеливания этапов просеивания и обработки матрицы. Для методов, пригодных для группы точек эллиптической кривой, применяются алгоритмы построения распределённой базы точек и поиска в ней.

Рассмотрено построение задачи дискретного логарифмирования (ДЛ) заданной размерности в $E(F_p)$. Проанализированы методы ДЛ, работающие в $E(F_p)$ и пригодные для распараллеливания, выделены общие вычислительно сложные участки, требующие разработки параллельных алгоритмов. Рассмотрены вопросы построения базы точек, распределённой между процессами, и организации взаимодействия процессов в схеме «полносвязный граф» по принципу однокругового турнира. Определены требования к абстрактному типу данных, представляющему точки эллиптической кривой, выбраны структуры данных и алгоритмы, используемые для представления участков базы данных. Разработаны параллельные алгоритмы, предназначенные для реализации ДЛ методами встречи посередине и встречи на случайном дереве, которые позволяют увеличить скорость решения задачи ДЛ за счёт использования распределённых вычислений. Описана возможность проведения предварительных вычислений с помощью рассмотренных методов, что ускоряет поиск конкретного логарифма при известных эллиптической кривой и образующей точке.

Экспериментальные данные, полученные с помощью реализованных программ, отражают эффективность разработанных алгоритмов. Для параллельных алгоритмов получены результаты, отражающие зависимость скорости вычислений от используемого числа процессоров и способа распределения данных.

Экспериментально показано, что для методов факторизации чисел и дискретного логарифмирования распараллеливание этапа построения БД выполняется с эффективностью, немного ниже линейной, причём с увеличением числа задействованных процессорных ядер эффективность падает. Это можно объяснить тем, что при увеличении числа вычислительных узлов растут затраты на разделение точек по диапазонам и на коммуникацию между узлами. Однако при 20-процессорных ядрах эффективность ещё достаточно высока (около 0,95). Распараллеливание алгоритмов на этапе сортировки показывает практически линейный рост производительности, что говорит об эффективности предложенных подходов распараллеливания.

Функции хэширования. Криптографической функцией хэширования (хэш-функцией) H называется отображение множества всех возможных сообщений (представленных в двоичном виде) во множество двоичных векторов конечной фиксированной длины n (множество хэш-значений, или хэш-кодов).

В 1989 г. Р. Меркль (Ralph C. Merkle) и И. Дамгорд (Ivan Damgaard) независимо предложили итеративный принцип построения криптографических функций хэширования. Данный принцип позволяет свести задачу построения хэш-функции на множестве сообщений различной длины к задаче построения отображения, действующего на множестве фиксированной конечной длины. По итеративному принципу построено абсолютное большинство хэш-функций, используемых в настоящее время на практике. Например, хэш-функции MD5, SHA-1, семейство хэш-функций SHA-2, отечественный стандарт на хэш-функцию ГОСТ Р 34.11-94.

Исходя из свойств криптографических хэш-функций, выделяют 3 типа атак:

1. Атака на обнаружение коллизий. Суть атаки состоит в нахождении двух произвольных сообщений m_1 и m_2, которые дают одинаковые хэш-значения $\text{hash}(m_1) = \text{hash}(m_2)$.

2. Атака нахождения первого прообраза. По известному хэш-значению h необходимо найти такое сообщение m, что hash(m) = h.

3. Атака нахождения второго прообраза. По данному сообщению m_1 необходимо найти отличное от него сообщение m_2 такое, что $\text{hash}(m_2) = \text{hash}(m_1)$. Данная атака, по сути, является вариантом атаки на обнаружение коллизий [44].

Для реализации данных атак применяются различные методы. Все методы криптографического анализа хэш-функций можно разделить на два класса:

- методы, не зависящие от алгоритма преобразования;
- методы, основанные на уязвимости алгоритма преобразования хэш-функции.

Коллективом кафедры БИТ ведутся разработки в области исследования стойкости современных функций хэширования к различным видам анализа. В частности, рассмотрены возможности использования метода дифференциального криптоанализа. Исследования проводились на упрощенной модели функции хэширования SHA [8]. В настоящий момент ведется исследование новой функции хэширования Skein, которая является финалистом конкурса SHA-3 [15].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Бабенко Л.К., Ищукова Е.А.* Современные алгоритмы блочного шифрования и методы их анализа. – М.: Гелиос АРВ, 2006. – С. 376.
2. *Шнайер Б.* Прикладная криптография: Протоколы, алгоритмы, исходные тексты на языке Си. – М.:ТРИУМФ, 2002. – С. 648.
3. *Biham E., Shamir A.* Differential Cryptanalysis of the Full 16-round DES, Crypto'92, Springer-Velgar, 1998. – P. 487.
4. *Babenko L.K., Ishchukova E.A.* Data Distribution Algoritms for Differential Cryptanalysis of DES // Proceeding of the Workshop on Computer Science and Information Technologies (CSIT'2007), Krasnousolsk, UFA, September 13-16, 2007. – Vol. 1. UFA State Aviation Technical University, 2007. – P. 198-201.
5. *Ищукова Е.А.* Применение рекурсивного алгоритма поиска в Б-деревьях для дифференциального криптоанализа алгоритма шифрования ГОСТ 28147-89 // Информационная безопасность. Ч. 2. – Таганрог: Изд-во: ТТИ ЮФУ, 2007. – С. 92-97.
6. *Babenko L.K., Ishchukova E.A.* Differential Analysis GOST Encryption Algorithm // Proceedings of the 3rd Internaternational Conference of Security of Information and Networks (SIN 2010), ACM. – New York, 2010. – P. 149-157.
7. *Бабенко Л.К. Ищукова Е.А.* Дифференциальный криптоанализ поточных шифров // Известия ЮФУ. Технические науки. – 2009. – № 11 (100). – С. 232-239.
8. *Бабенко Л.К. Ищукова Е.А.* Дифференциальный криптоанализ упрощенной функции хэширования SHA // Известия ЮФУ. Технические науки. – 2010. – № 11 (112). – С. 99-106.
9. *Бабенко Л.К., Ищукова Е.А.* Анализ современных криптографических систем с помощью метода дифференциального криптоанализа // Актуальные аспекты защиты информации в Южном федеральном университете. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – С. 102-181.
10. *Courtois N., Klimov A., Patarin J., Shamir A.* Efficient algorithms for solving overdefined systems of multivariate polynomial equations // EUROCRYPT, 2000. – P. 392-407.
11. *Babenko L.K., Ishcukova E.A., Maro E.A.* Algebraic analysis of GOST encryption algorithm // SIN'11 Proceedings of the 4th International Conference of Security of Information and Networks. – 2011 Sidney, Australia. – P. 115-123.
12. *Бабенко Л.К., Маро Е.А* Алгебраический анализ современных систем защиты информации // Актуальные аспекты защиты информации в Южном федеральном университете. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – С. 181-207.
13. *Babenko, L., Sidorov I.* Parallel algorithms for discrete log solving and their effectiveness // Proceedings of the Workshop on Computer Science and Information Technologies (CSIT'2009), Crete, Greece, October 5-8, 2009. Vol. 2. Ufa State Avation Technical University, 2009. – P. 217-222.
14. *Бабенко Л.К., Сидоров И.Д.* Параллельные алгоритмы криптоанализа асимметричных систем // Актуальные аспекты защиты информации в Южном федеральном университете. – Таганрог: Изд-во ТТИ ЮФУ, 2011. – С. 207-252.
15. *Бабенко Л.К., Ищукова Е.А., Щеткина Е.А.* Новая функция хэширования Skein и подходы к ее анализу // Моделирование устойчивого регионального развития. – Нальчик: Изд-во КБНЦ РАН, 2011. – С. 78-84.

Статью рекомендовал к опубликованию д.т.н., профессор Я.Е. Ромм.

Бабенко Людмила Климентьевна – Технологический институт федерального государственного автономного образовательного учреждения высшего профессионального образования «Южный федеральный университет» в г. Таганроге; e-mail: blk@fib.tsure.ru; 347928, г. Таганрог, ул. Чехова, 2; тел.: 88634312018; кафедра безопасности информационных технологий; д.т.н.; профессор.

Ищукова Евгения Александровна – e-mail: jekky82@mail.ru; тел.: 88634371905; кафедра безопасности информационных технологий; к.т.н.; доцент.

Маро Екатерина Александровна – e-mail: marokat@gmail.com; тел.: 88634371905; кафедра безопасности информационных технологий; ассистент.

Сидоров Игорь Дмитриевич – e-mail: idsidorov@gmail.com; тел.: 88634371905; кафедра безопасности информационных технологий; к.т.н.; доцент.

Кравченко Павел Павлович – e-mail: kravch@tsure.ru; 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371673; кафедра математического обеспечения и применения ЭВМ; зав. кафедрой.

Babenko Lyudmila Klimentevna – Taganrog Institute of Technology – Federal State-Owned Autonomy Educational Establishment of Higher Vocational Education “Southern Federal University”; e-mail: blk@fib.tsure.ru; 2, Chekhov street, Taganrog, 347928, Russia; phone: +78634312018; the department of security in data processing technologies; dr. of eng. sc.; professor.

Ischukova Evgeniya Aleksandrovna – e-mail: jekky82@mail.ru; phone: +78634371905; the department of security in data processing technologies; cand. of eng. sc.; associate professor.

Maro Ekaterina Aleksandrovna – e-mail: marokat@gmail.com; phone: +78634371905; the department of security in data processing technologies; assistant.

Sidorov Igor Dmitrievich – e-mail: idsidorov@gmail.com; phone: +78634371905; the department of security in data processing technologies; cand. of eng. sc.; associate professor.

Kravchenko Pavel Pavlovich – e-mail: kravch@tsure.ru; 44, Nekrasovskyi, Taganrog, 347928, Russia; phone: +78634371673; the department of software engineering; head of the department.

УДК 004.491

Л.К. Бабенко, Е.П. Тумоян, К.В. Цыганок, М.В. Аникеев

КЛАССИФИКАЦИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ ПОВЕДЕНЧЕСКИХ ПРИЗНАКОВ

Классификация является распространенной задачей при анализе вредоносного программного обеспечения и генерации сигнатур для него. Обычным способом классификации вредоносного кода является экспертная оценка похожести антивирусных образцов квалифицированным вирусным аналитиком.

В данной работе предлагается новый метод классификации вредоносного кода на основе поведенческих признаков – последовательности вызовов WinAPI и их аргументов, а также файлов, создаваемых анализируемым приложением. Метод обеспечивает получение двумерного вектора, характеризующего данную программу. Наборы характеризующих векторов кластеризуются с использованием оригинального алгоритма нечеткой кластеризации. Полученные кластеры отражают группы программ, демонстрирующие сходную, с поведенческой точки зрения, активность. Метод был экспериментально исследован на исполняемых тестовых файлах, защищенных упаковкой и шифрованием, а также реальных образцах вредоносного программного обеспечения.

Обнаружение вредоносных программ; метаморфные преобразования; кластеризация; компьютерная безопасность.