ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

УДК 004.414

Пороговое разделение файлов на основе битовых масок: идея и возможное применение

Н. С. Могилевская
(Донской государственный технический университет),
Р. В. Кульбикаян
(Ростовский государственный университет путей сообщения),
Л. А. Журавлёв
(Донской государственный технический университет)

Предлагается новый метод порогового разделения файла любого формата на n частей таким образом, чтобы для его корректного восстановления было необходимо собрать не менее k(<n) частей. Предложенный метод может быть использован для децентрализованного хранения файлов, для передачи файлов по многоканальным сетям, а также для защиты от несанкционированного доступа к информации, содержащейся в файле.
Ключевые слова: *пороговое разделение секрета, метод битовых масок, безопасность файлов, децентрализованное хранение файлов, передача файла по многоканальной системе связи.*

Введение. Идея данной работы родилась на стыке трёх задач, для решения которых в том или ином виде используется разделение данных на части для повышения уровня их сохранности. Так, первая задача состоит в предохранении секретной информации (ключей) от потери, разделении ответственности за принятие решения и предотвращении атак, связанных с человеческим фактором, таких, как подкуп, шантаж, захват людей, имеющих отношение к секретной информации. Решается эта задача с помощью пороговых схем разделения секрета, разработанных в теории криптографических протоколов. (k, n)-пороговым протоколом разделения секрета называют распределённый алгоритм, в котором некоторый числовой секрет N разделяется на n частей-долей и распределяется между участниками таким образом, чтобы любые k участников, собравшись вместе, могли восстановить секрет N, а любые $(k - 1)$ участников ничего не могли узнать о секрете [1, 2, 3]. На сегодняшний день существует большое количество схем разделения секрета, например [1, 3]. Наиболее известной, пожалуй, является (k, n)-пороговая схема Ади Шамира, в основе которой лежит известный алгебраический факт, что для восстановления всех коэффициентов полинома $f(x)$ степени $k - 1$ необходимо знать значение $f(x)$ в k различных точках. Согласно схеме Шамира, используются полиномиальные уравнения в конечном поле F_p, где p — простое число, больше количества возможных долей n и больше любого возможного секрета [3]. К подготовительной части этой схемы относится генерация полинома $f(x)$ степени $k - 1$ со случайными коэффициентами из F_p, такого, что значение секрета равно $f(0)$. Долей секрета участника j ($j = 1, \ldots, n$) схемы является пара вида $(x_j, f(x_j))$, где $x_j = 1, \ldots, p - 1$. Для восстановления секрета $f(0)$, согласно (k, n)-пороговой схеме Шамира, используется интерполяционная формула Лагранжа. Ещё одна популярная схема предложена Джорджем Блэкли [3], в которой секретом является одна из координат точки Q в k-мерном пространстве, а долями секрета являются уравнения плоскостей, пересекающихся в Q. Для восстановления секрета не-

обходимо решить систему, состоящую из k уравнений плоскостей, которые являются легальными долями секрета.

Вторая задача, в которой используется разделение исходного файла на части, подробно описана в работе [4], где предложена схема организации децентрализованного отказоустойчивого хранилища данных. Точнее, на основании представления исходного файла как множества векторов из элементов полей Галуа предложен метод разделения файла на n частей таким образом, что по любым $k < n$ из них можно восстановить исходный файл. Различные части исходного файла предлагается хранить на различных серверах таким образом, чтобы в любой момент можно было получить исходный файл, даже в случае отказа какого-либо из серверов.

В третьей задаче авторами [5] предлагается вносить в данные изменения, которые их «портят», лишая смысла. А именно: в исходный файл предлагается внести «ущерб», точнее, уменьшить длины кодов букв за пределами их информационной неизбыточности, т. е. внести изменения, искажающие смысл исходного сообщения. Так, например, битовым последовательностям фиксированной длины ставятся в соответствие последовательности, возможно, меньшей и непостоянной длины. Таким образом, исходный файл разделяется на три неравнозначных доли: ущербный файл, «ущерб» и таблица замен. Для восстановления исходного файла необходимы все три части. Новые доли предполагается передавать между участниками информационной системы по разным каналам связи, что уменьшает вероятность одновременного перехвата злоумышленником всех трёх частей, следовательно, уменьшает и вероятность несанкционированного доступа к данным, содержащимся в исходном файле.

Очевидно, что было бы полезным иметь общий механизм разделения исходных данных на части, который может быть применён для решения всех трёх типов указанных задач. Такой механизм разработан авторами данной статьи и назван алгоритмом битовых масок.

Далее будет описана идея предлагаемого метода, алгоритм построения битовых масок, возможные области его применения и приведён пример протокола разделения файла с использованием предложенного метода.

Основная идея предлагаемого метода порогового разделения файлов. Кратко идея предлагаемого метода состоит в том, что исходный файл, назовём его секретом, разделяется на n долей таким образом, чтобы для восстановления секрета было необходимо объединить не менее k долей, где $k < n$.

Рассмотрим исходный файл как последовательность битовых отрезков $\{S_1, S_2, S_3, S_4, S_5, S_6, \ldots\}$ некоторой фиксированной длины S, назовём эти отрезки сегментами. В зависимости от исходного файла и параметров системы в качестве сегмента может быть принят, например, 1 бит или 1 байт, или группа байтов, группа пикселов графического файла или несколько отсчётов аудиофайла, т. е. любой объём данных, удобный для обработки. Для формирования каждой доли генерируется уникальная битовая маска m_i, $i = 1, \ldots, n$, которая циклически применяется к секрету таким образом, что каждому сегменту секрета соответствует один бит маски. Если текущий бит маски нулевой, то соответствующий сегмент секрета отбрасывается, а если бит маски единичный, то соответствующий сегмент записывается в долю. Графически описанная идея представлена на рис. 1. Фактически применяется логическая операция И, аргументами которой являются бит маски и сегмент секрета, таким образом, долей секрета является новый файл, частично содержащий исходный. Очевидно, что размер построенной доли меньше размера исходного файла. Для оценки размера доли D необходимо знать размер исходного файла N, используемый размер сегмента S и число нулей α и единиц β в маске.

$$\left\lfloor \frac{N}{S \cdot (\alpha + \beta)} \right\rfloor \cdot \beta \cdot S \le D \le \left\lceil \frac{N}{S \cdot (\alpha + \beta)} \right\rceil \cdot \beta \cdot S \qquad (1)$$

где $\lfloor\ \rfloor, \lceil\ \rceil$ используются для обозначения операции округления до ближайшего меньшего и ближайшего большего соответственно, а $\alpha + \beta = C_n^{k-1}$.

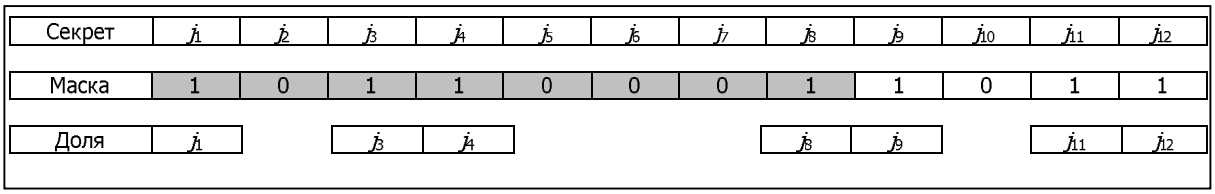

Рис. 1. Визуализация генерации доли секрета путём наложения битовой маски на исходный файл. Маска 10110001 применяется циклически

Количество нулей и единиц в маске определяется следующими соображениями. Очевидно, для того чтобы реализовать пороговое восстановление секрета, необходимо, чтобы маски k различных долей при применении к ним побитового логического ИЛИ давали в результате вектор, содержащий только единицы. Рассмотрим пример масок для случая $n = 4$, $k = 3$.

Маска 1: 0 0 0 1 1 1
Маска 2: 0 1 1 0 0 1
Маска 3: 1 0 1 0 1 0
Маска 4: 1 1 0 1 0 0

Легко проверить, что, применяя логическое ИЛИ к трём или более любым маскам, получаем в результате вектор, состоящий из одних только единиц. Длина масок определяется значением C_n^{k-1}, число нулей и единиц у всех масок постоянное.

Алгоритм генерации масок. Представим алгоритм генерации масок для заданного числа участников n и порогового значения k.

Шаг 1. Составим матрицу, строками которой будут все возможные векторы длины n, содержащие $(k - 1)$ нулевой элемент и $(n - k + 1)$ единичных элементов.

Шаг 2. Транспонируем полученную матрицу, очевидно, что её размеры будут $n \times C_n^{k-1}$. Каждая строка этой матрицы будет представлять отдельную маску для каждой из n долей.

На рис. 2 представлен пример работы алгоритма генерации масок для параметров $n = 5$, $k = 3$. Очевидно, что предложенному алгоритму свойственна масштабируемость, т. е. при необходимости можно увеличить число участников реализуемой системы порогового разделения файла. Так, если известно, что число участников разделения файла может быть увеличено, то при формировании долей можно выбрать число n заведомо большее, чем необходимо в момент формирования долей, тогда в будущем можно будет увеличивать число участников, раздавая им доли, сгенерированные с ещё не использованными масками.

$$C_n^{k-1} = \frac{n!}{(k-1)!(n-k-1)!}$$

$$C_5^2 = \frac{5!}{2!\cdot 3!} = 10$$

Шаг 1

1) 1 0 0 0 1
2) 1 0 0 1 0
3) 1 0 1 0 0
4) 1 1 0 0 0
5) 0 1 1 0 0
6) 0 1 0 1 0
7) 0 1 0 0 1
8) 0 0 1 1 0
9) 0 0 1 0 1
10) 0 0 0 1 1

Шаг 2

Маска 1: 1111000000
Маска 2: 0001111000
Маска 3: 0010100110
Маска 4: 0100010101
Маска 5: 1000001011

Рис. 2. Пример работы алгоритма генерации масок для параметров $n = 5$, $k = 3$

По примеру, приведённому на рис. 1, и (1) видно, что размер каждой доли будет меньше размера исходного файла, однако размер k долей будет превышать размер секрета. Этот факт не является недостатком, такая же ситуация справедлива и для протоколов разделения секрета, и для теории ущербных текстов. Разница между размером доли и размером секрета варьируется в достаточно широком диапазоне, в зависимости от используемых параметров. Пример связи между размером доли и размером файла-секрета можно оценить по следующей таблице, построенной аналитически, для общего числа участников разделения файла $n = 4$ и $n = 6$.

Оценка уменьшения размера доли по сравнению с размером секрета

Пороговое значение k	Длина маски $\alpha + \beta$	Число нулей в маске α	Разница в размере доли и секрета, %
$n = 4$			
2	4	1	25
3	6	2	50
4	4	3	75
$n = 6$			
2	6	1	15
3	15	5	34
4	20	10	50
5	15	10	67
6	6	5	83

Возможные приложения. Рассмотрим несколько различных вариантов применения предложенного метода.

1. Предложенный метод порогового разделения файлов может быть использован как вариант RAID-технологий как для локального, так и для распределённого хранения и восстановления данных. Напомним, что аббревиатура RAID расшифровывается как Redundant Array of Independent Disks — «отказоустойчивый массив из независимых дисков» и представляет собой концепцию структуры, состоящей из нескольких дисков, объединённых в группу, и обеспечивающей отказоустойчивость. В такой системе каждый файл предлагается хранить в виде некоторого набора частей, количество которых может меняться во времени. Всегда в любой момент времени для существующих в системе n частей выполняется условие, что из любых k кусков можно полностью собрать файл.

Хранение данных можно организовать таким образом: n долей файла распределить по n серверам либо поместить на один сервер k долей — так, чтобы всё необходимое для восстановления файла можно было получить с одного сервера, а на остальные k серверов разместить доли только для осуществления сборки в случае недоступности первого сервера. Кроме этого, при распределении долей можно учитывать загруженность серверов, например, размещать больше долей файла на серверы с меньшей загрузкой.

2. Предложенный метод разделения файлов может быть использован для передачи файлов по многоканальным системам связи с целью повышения скорости передачи. В этом случае данный метод необходимо либо снабдить помехоустойчивой защитой, либо передавать более k долей, чтобы приёмник не только получал файлы быстро, но так же быстро и безошибочно мог восстановить исходный файл.

3. В случае разделения файла из-за соображений безопасности между группой участников или между различными линиями связи желательно снабдить данный метод надёжным алгоритмом шифрования и решить вопрос о безопасном распределении ключей шифрования и долей между участниками информационной системы.

Какое бы применение предложенного метода ни было бы выбрано, очевидно, что необходимо хранить не только долю секрета, но и маску, а также и некоторые другие параметры

протокола порогового разделения файлов. Рассмотрим один из примеров применения предложенного метода.

Пример применения алгоритма разделения файла на основе метода битовых масок для защиты от несанкционированного доступа. Очевидно, что перед началом работы протокола все участники должны знать последовательность шагов протокола и быть готовыми ему следовать, также должен быть определён используемый алгоритм шифрования.

Этап разделения секрета. Подготовительные шаги.

Шаг 1. Зафиксируем параметры схемы разделения секрета: n — максимально возможное количество участников системы, k — пороговое значение, E — используемый криптографический алгоритм. Сконструируем заголовок для каждой из долей секрета по следующей схеме: первое поле — размер сегмента S, второе поле — размер секрета N.

Шаг 2. Сгенерируем n битовых масок по предложенному выше алгоритму с параметрами n и k.

Этап разделения секрета. Основная часть.

Для каждой i-й доли выполнять шаги 3—7.

Шаг 3. Обработаем заголовок маской побитовой логической операцией И: если в маске ноль, то бит заголовка тоже ноль, иначе оставляем бит данных.

Шаг 4. Циклически применим маску к секрету, используем логическое И для каждого бита маски и соответствующего сегмента секрета.

Шаг 5. Конкатенируем заголовок из шага 3 и обработанный маской секрет из шага 4. Получим файл f_i.

Шаг 6. Зашифруем файл f_i алгоритмом E и секретным ключом K_i i-го пользователя. Заметим, что в качестве шифрования можно выбрать любой известный стойкий алгоритм шифрования, использовать модульное умножение, перестановку бит или любой другой подходящий способ.

Шаг 7. Сформируем долю секрета как пару $(m_i, E_K(f_i))$, где m_i — маска i-го участника, $E_K(f_i)$ — зашифрованный файл, содержащий долю, полученную из секрета.

Этап восстановления секрета.

Шаг 1. Соберём k различных долей секрета $(m_i, E_K(f_i))$. Далее владелец каждой доли снимает шифрование со своей доли, получаем $\{(m_{i2}, f_{i1}), (m_{i2}, f_{i2}), \ldots, (m_{ik}, f_{ik})\}$.

Шаг 2. Для каждой доли отделим заголовок и секретную часть. Получим последовательность заголовков $\{header_{j1}, header_{j2}, \ldots, header_{jk}\}$ и последовательность долей $\{Share_{j1}, Share_{j2}, \ldots, Share_{jk}\}$.

Шаг 3. Применим операцию побитового логического ИЛИ ко всем заголовкам. В результате получим заголовок, из которого восстанавливаем размер сегмента S и длину файла-секрета N.

Шаг 4. Восстанавливаем исходный файл-секрет следующим образом. Используем долю (m_{j1}, f_{j1}) и сформируем шаблон F' файла-секрета. Для этого циклически используем маску m_{j1} и для каждого её единичного элемента записываем в шаблон соответствующий сегмент из f_{j1}, а для каждого нулевого элемента маски записываем в шаблон сегмент, заполненный нулями, количество сегментов определяется хранящейся в заголовке длиной файла-секрета.

Шаг 5. Оставшиеся $(k - 1)$ доли используем для восстановления файла-секрета следующим образом: циклически используем маску m_j $(j = 2, 3, \ldots, k)$ и для каждого её единичного элемента записываем в шаблон F' соответствующий сегмент из f_j, а в случае нулевого элемента маски пропускаем в шаблоне соответствующий сегмент. В результате получаем файл F', возможно, совпадающий с оригинальным файлом-секретом.

Шаг 8. Сравниваем длину F' и длину файла-секрета N. Если эти величины не совпадают, то уменьшаем длину файла F', обрезая «лишнее» с конца. Заметим, что несовпадение длин может появиться в случае, когда размер сегмента не 1 бит и не кратен 1 байту.

Замечание. В предложенном примере использования метода побитовых масок доли секрета формировались в виде пары (m_{j1}, f_{j1}). Несложно видоизменить долю секрета таким образом, чтобы пользователю не пришлось хранить значение маски. А именно: внести в заголовок доли значения n и k, а в алгоритме генерации масок зафиксировать способ построения строк на шаге 1 таким образом, чтобы в различных случаях запуска этого алгоритма с фиксированными параметрами результирующая матрица получалась постоянной. Тогда доля секрета будет выглядеть как пара (i, f_i), где i — номер доли.

Выводы. Предложен метод битовых масок, который может быть использован в задачах разделения файлов для организации децентрализованного отказоустойчивого хранилища данных, для разделения файлов в системах многоканальной связи или для порогового разделения секретной информации между группой пользователей. Приведён пример использования предложенного метода в алгоритме разделения файла для защиты от несанкционированного доступа. Дальнейшие направления данной работы связаны с построением, экспериментальным исследованием и оценкой протоколов применения разработанного метода битовых масок в различных областях.

Библиографический список

1. Шнайер, Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. — Москва: Триумф, 2002. — 816 с.

2. Черёмушкин, А. В. Криптографические протоколы: основные свойства и уязвимости / А. В. Черёмушкин. — Москва: Ин-т криптографии, 2009. — 272 с.

3. Могилевская, Н. С. Методы порогового разделения секрета. Схема Блэкли. Схема Шамира. Метод. указания по курсу «Криптографические протоколы» / Н. С. Могилевская. — Ростов-на-Дону: Изд. центр ДГТУ, 2011. — 12 с.

4. Тормасов, А. Г. Математическое моделирование средств управления ресурсами и данными в распределённых и виртуализованных средах: автореф. … д-ра физ.-мат. наук: 05.13.18 / А. Г. Тормасов. — Москва, 2008. — 38 с.

5. Мищенко, В. А. Ущербные тексты и многоканальная криптография / В. А. Мищенко, Ю. В. Виланский. — Минск: Энциклопедикс, 2007. — 292 с.

Материал поступил в редакцию 02.12.2011.

References

1. Shnajer, B. Prikladnaya kriptografiya. Protokoly`, algoritmy`, isxodny`e teksty` na yazy`ke Si / B. Shnajer. — Moskva: Triumf, 2002. — 816 s. — In Russian.

2. Cheryomushkin, A. V. Kriptograficheskie protokoly`: osnovny`e svojstva i uyazvimosti / A. V. Cheryomushkin. — Moskva: In-t kriptografii, 2009. — 272 s. — In Russian.

3. Mogilevskaya, N. S. Metody` porogovogo razdeleniya sekreta. Sxema Ble`kli. Sxema Shamira. Metod. ukazaniya po kursu «Kriptograficheskie protokoly`» / N. S. Mogilevskaya. — Rostov-na-Donu: Izd. centr DGTU, 2011. — 12 s. — In Russian.

4. Tormasov, A. G. Matematicheskoe modelirovanie sredstv upravleniya resursami i danny`mi v raspredelyonny`x i virtualizovanny`x sredax: avtoref. … d-ra fiz.-mat. nauk: 05.13.18 / A. G. Tormasov. — Moskva, 2008. — 38 s. — In Russian.

5. Mishhenko, V. A. Ushherbny`e teksty` i mnogokanal`naya kriptografiya / V. A. Mishhenko, Yu. V. Vilanskij. — Minsk: E`nciklopediks, 2007. — 292 s. — In Russian.

THRESHOLD FILE SHARING BASED ON BIT MASKS: CONCEPT AND POSSIBLE USE

N. S. Mogilevskaya
(Don State Technical University),
R. V. Kulbikayan
(Rostov State Transport University),
L. A. Zhuravlev
(Don State Technical University)

A new method of the threshold sharing of any format file on n parts in such a way that it would be necessary to collect at least k(<n) parts for its correct recovery is offered. The proposed method can be used for decentralized filing, file transfer through the multichannel networks, and for unauthorized access protection of the information contained in the file.
Keywords: *threshold secret sharing, bit mask method, file security, decentralized filing, file transfer through multichannel networks.*