УДК 004.056.55

Р.Т. Файзуллин, Д.А. Сагайдак

Приложение алгоритма префиксного кодирования массива данных в схеме разделения секрета потока видеоданных

Предложен вариант алгоритма, использующего префиксное кодирование массива данных и основанного на примитивизации дельта-кода Элиаса, в схеме разделения секрета. Описана возможность использования предложенного алгоритма в схеме разделения видеопотока данных. Выполнена программная реализация предложенного метода в среде программирования Borland C++.

Ключевые слова: разделение секрета, шардинг, дельта-код Элиаса, разделение видеопотока, представление в формате RGB.

На современном этапе развития информационных технологий одними из самых актуальных задач являются задачи безопасного хранения и передачи информации. Раскрытие конфиденциальных данных или их безвозвратная потеря могут привести к фатальным последствиям. Поэтому большинство организаций (пользователей) стремятся защитить свою конфиденциальную информацию, зачастую используя криптографические алгоритмы, которые порой могут оказаться недостаточно стойкими или недостаточно быстрыми для реализации поставленной задачи. Так же использование криптографических алгоритмов может требовать специальных навыков пользователей и наличия вычислительных ресурсов. Тем самым встает вопрос о необходимости использования алгоритмов, обеспечивающих защиту конфиденциальной информации, которые способны осуществлять быстрое и стойкое преобразование и не требовать от пользователя какой-либо специальной подготовки.

Зачастую для реализации поставленных целей прибегают к использованию оконечных аппаратных комплексов, которые осуществляют криптографическое преобразование проходящего через них потока, типичными примерами таких устройств могут являться устройства, осуществляющие преобразование видеосигнала, где без знания специального ключа нельзя дешифровать преобразуемый ими видеосигнал. Но при использовании таких устройств пользователи могут столкнуться с рядом проблем, таких как маленькая длина ключа или его слабая стойкость, шифрование всего видеопотока одним ключом, проблемы синхронизации при использовании медленных алгоритмов шифрования и т.п. Кроме того, высокая коррелированность видеоданных и способность человека к распознаванию зашумленных изображений позволяют легко, «на лету», восстанавливать данные. Таким образом, все так же является актуальным использование алгоритмов, не требующих значительных вычислительных ресурсов и осуществляющих преобразование с использованием периодически меняющегося ключа, в таких случаях можно прибегнуть к схеме разделения секрета. Принципиальным решением проблемы может быть применение доказуемо стойких схем с разделением секрета, где существенно большая часть секрета передается по открытому каналу, а меньшая или шифруется, или иначе, передается по защищенному каналу передачи данных. Под меньшей следует понимать ту часть секрета, длина битовой записи которой оценивается логарифмом от длины записи большей части. В настоящей работе предлагаются схема кодирования и эффективное преобразование данных, которое можно рассматривать как доказуемо стойкую схему разделения секрета на существенно неравные части.

Постановка задачи

Рассмотрим задачу хранения большого числа массивов данных, длины записи которых существенно различаются. Пусть даны n битовых векторов $A_1,...,A_n$, размерности которых равны $M_1,...,M_n$ и дисперсии M_i распределены равномерно в достаточно большом интервале.

В этом случае возникает проблема экономичной записи данных, которая в настоящее время решается различными способами: шардингом [1], т.е. грубым физическим разделением данных по различным носителям данных, введением различных типов данных, наподобие CHAR и VARCHAR, разделением данных маркерами. Но если M_i варьируются от 10^3 до 10^{10}, а n изменяется, то отве-

дение равных областей памяти для каждого A_i исключительно неэффективно, а разделение данных специальной строкой бит (маркером) неэффективно по времени поиска этого маркера, и нет никакой гарантии, что выбранная в качестве маркера строка не встречается ни в одном из A_i.

Использование алгоритма, представляющего собой примитивизацию дельта-кода Элиаса

Рассмотри алгоритм, представляющий собой примитивизацию дельта-кода Элиаса (универсальный код для кодирования целых чисел, разработанный П. Элиасом) [2], который позволяет избежать указанных трудностей.

Первые l бит заполним нулями, где l – это длина записи числа n, далее идёт сама запись числа n, например, пусть даны $n=3$ битовых векторов, тогда запись числа n в двоичной системе счисления равна 11 и $l=00$. И на первом этапе получается следующая числовая последовательность: 0011. Далее, m_i бит заполняются нулями, где m_i – число бит, необходимых для записи длины вектора A_i в двоичной системе счисления. Например, в предыдущем примере $n=3$, следовательно, имеется три битовых вектора A_1, A_2, A_3. Пусть $A_1=111011$, $A_2=10111$, $A_3=101$, тогда размерности этих битовых последовательностей равны $M_1=6$, $M_2=5$, $M_3=3$ соответственно, а $m_1=000$, $m_2=000$, $m_3=00$. Тем самым на втором этапе получится следующая последовательность: 0001100001010011. Третьим этапом формирования последовательности является последовательная запись самих векторов A_1, A_2, A_3. Например, для приведённых выше примеров получится следующая исходная последовательность: 00110001100001010011111101110111101.

Обратим внимание на то, что, зная диапазон возможных изменений A_i, можно записывать A_i в m_i+d_i позициях, предваряя или дополняя нулями значащие цифры A_i. Это позволяет легко перезаписывать и дописывать массивы и их новые значения, не усложняя структуру данных.

Предполагается возможным использование предложенного выше метода как основы схемы разделения секрета для видеопотока данных, и мы попытаемся построить алгоритм, не требующий значительных вычислительных ресурсов.

Пусть имеется поток видеоинформации, передаваемый по каналам связи, осуществляется разбиение данного видеопотока на фреймы. Производится построчное чтение пикселей фрейма, затем для каждого пикселя строки находятся его значения в формате RGB (red, green, blue) в двоичной системе счисления (размерностью 24 бита, т.е. по 8 бит для каждого цвета) и записываются последовательно друг за другом в одну строку, создавая последовательность, состоящую из нулей и единиц. Каждая такая достаточно длинная последовательность строки прочитанных пикселей разбивается на n битовых векторов $A_1,A_2,...,A_n$, разных размерностей $M_1,...,M_n$. Затем все полученные строки, состоящие из префикса и зашифрованных или преобразованных $A_1,A_2,...,A_n \Rightarrow C_1,..,C_n$ (без изменения длин записи), объединяются в одну битовую строку. Очевидно, что, не зная префиксов, определение границ разделения сводится к переборной задаче.

Простое шифрование является затратной по времени операцией, и поэтому предлагается модификация с наиболее эффективным по времени преобразованием.

Пусть имеется поток определенного («телевизионного») формата 720×576 пикселей 25 кадров в секунду в формате RGB (в дальнейшем будет осуществляться преобразование видеопотоков стандартных форматов: 720×576, 640×480, 352×288 (CIF – Common Interchenge Format), 176×144 (QCIF – Quartered Common Interchenge Format)), т.е. размерность изображения является известной и выбирается из одного из стандартов. Здесь осуществляются аналогичные действия: осуществляется разбиение видеопотока на фреймы, для каждого пикселя фрейма находятся его значения в формате RGB (red, green, blue) в двоичной системе счисления и записываются последовательно друг за другом (изображение считывается построчно слева направо), затем полученная битовая последовательность (битовая последовательность состоит из последовательно записанных друг за другом значения пикселей строк в двоичной системе счисления) разбивается на n случайных битовых векторов разной размерности. Формируется префикс с указанием, на сколько n частей разбита последовательность, и с указанием длин каждых полученных векторов $A_1,A_2,...,A_n$.

В качестве генератора случайных чисел в программе, реализующей описываемый метод (произвольно генерируются размерности векторов $A_1,A_2,...,A_n$ размером от 500 до 1000 бит), используется генератор псевдослучайных чисел RandomRange(), встроенный в среду программирования

Borland (C++, Delphi) и удовлетворяющий набору тестов, определённому стандартом FIPS 140-1 (Federal Information Processing Standards) [3].

Над полученными битовыми векторами $A_1, A_2, ..., A_n$ осуществляются следующие операции: $X_1 = A_1$, $X_2 = A_1 + A_2$, …, $X_n = A_{n-1} + A_n$ (где «+» – побитовое сложение по модулю 2), полученные битовые векторы $X_1, X_2, ..., X_n$ записываются последовательно друг за другом и дописываются к сформированному выше префиксу.

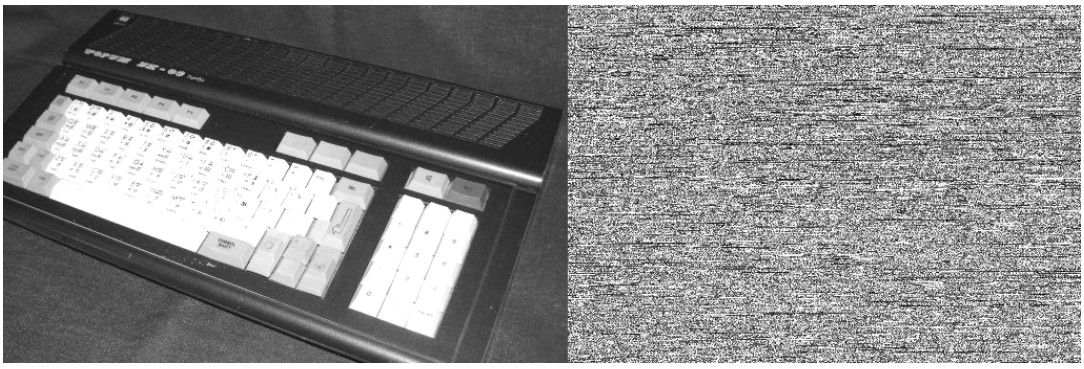
Также возможно, что для получения битовых векторов $X_1, X_2, ..., X_n$ вместо операции «+» – побитовое сложение по модулю 2, описанной выше, можно воспользоваться одним из режимов шифрования (метод применения блочного шифра, позволяющий преобразовать последовательность блоков открытых данных в последовательность блоков зашифрованных данных), таких как CBC (Cipher Block Chaining), CFB (режим гаммирования с обратной связью, Cipher Feedback).

Тем самым, если атакующему станет известна последовательность, состоящая из последовательно записанных векторов, он не сможет восстановить исходную последовательность без знания сформированного префикса. Если осуществлять посылку основной битовой последовательности и префикса по различным каналам связи, то будет обеспечиваться должный уровень конфиденциальности передаваемой информации.

В случае если атакующий попытается восстановить исходное изображение фрейма, зная исходные размеры изображения и только битовую последовательность, без полученного префикса, в результате получим, например, следующие изображения (рис. 1–3).

а *б*

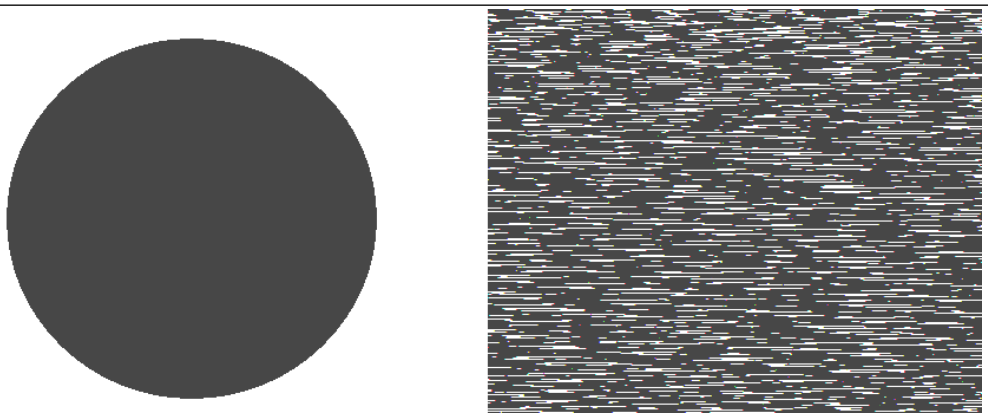
Рис. 1. Исходное изображение – *а*; изображение, полученное атакующим при попытке восстановить изображение из имеющейся у него последовательности бит, – *б*



а *б*

Рис. 2. Исходное изображение – *а*; изображение, полученное атакующим при попытке восстановить изображение из имеющейся у него последовательности бит, – *б*

Как видно из полученных изображений (см. рис. 1–3), даже зная полученную последовательность и применяя всякого рода перестановки, атакующему все равно не удастся восстановить исходные изображения без знания размеров битовых векторов, полученных в результате разбиения на произвольное число частей исходной битовой последовательности.



а *б*

Рис. 3. Исходное изображение – *а*; изображение, полученное атакующим при попытке восстановить изображение из имеющейся у него последовательности бит, – *б*

Понятно, что для каждого фрейма видеопотока формируется свой префикс – «ключевая последовательность», что позволяет говорить об аналогии с преобразованием, где используется одноразовый ключ.

При наличии у пользователя информации о размере изображения (размер «стандартный» и известен всем), сформированного префикса и битовой последовательности, ему удастся восстановить исходное изображение.

Но даже если атакующему станет известна часть префикса, но не A_1, и данные будут разделены на равные части $A_1, A_2, ..., A_n$, ему все равно не удастся восстановить исходное изображения, т.к. задача сводится к решению неопределенной системы уравнений из n уравнений с $n+1$ неизвестными:

$$A_1 + A_2 = X_2,$$

$$\ldots..$$

$$A_{n-1} + A_n = X_n.$$

Восстановить изображение можно только подбором бит, но в случае, когда длина записи X_1 больше, чем 80 бит, задача становится принципиально не решаемой, т.к. осуществить перебор на имеющейся в данное время вычислительной технике невозможно.

Описанный метод не требует значительных вычислительных ресурсов и способен осуществлять преобразование данных «на лету».

Заключение

1. Предложена схема разделения секрета, основанная на примитивизации дельта-кода Элиаса, описана его теоретическая часть.

2. Подготовлена программная реализация, осуществляющая выполнение описанного выше алгоритма, по полученным результатам программной реализации сделаны выводы об эффективности использования предложенного метода для сокрытия конфиденциальной информации.

Литература

1. Rahul Roy (July 28, 2008). Shard – A Database Design [Электронный ресурс]. – Режим доступа: http://technoroy.blogspot.com/2008/07/shard-database-design.html, свободный (дата обращения: 29.04.2012).

2. Elias P. Universal codeword sets and representations of the integers // IEEE Transactions on Information Theory. – 1975. – Vol. 21, issue 2, № 2. – P. 194–203.

3. Federal Information Processing Standards Publication. FIPS PUB 140-1. Security Requirements for Cryptographic Modules. – U.S., Department of commerce, National institute of standards and technology, 1994. – 53 с.

Файзуллин Рашит Тагирович
Д-р техн. наук., проф. каф. средств связи и информационной безопасности
Омского государственного технического университета (ОмГТУ)
Тел.: (3812) 62-87-07
Эл. почта: frt@omgtu.ru

Сагайдак Дмитрий Анатольевич
Аспирант каф. комплексной защиты информации ОмГТУ
Тел.: 8 (950) 780-27-77
Эл. почта: sagaydak.dmitriy@gmail.com

Faizullin R.T., Sagaydak D.A.
The application of the algorithm prefix coding scheme of the array data in secret sharing video stream

A variant of the algorithm that uses a prefix encoding of the array data based on delta-primitivization Elias code, the secret sharing scheme.
Keywords: separation of the secret, sharding, Elias delta code.