ОБРАБОТКА ИЗОБРАЖЕНИЙ

СХЕМА РАЗДЕЛЕНИЯ СЕКРЕТА НА ОСНОВЕ МЕТРИЧЕСКИХ ХАРАКТЕРИСТИК ДАННЫХ ДЛЯ ЗАЩИЩЕННОЙ ПЕРЕДАЧИ ВИДЕОПОТОКОВ

А.А. Свенч[1], Р.Т. Файзуллин[1]
[1]Омский Государственный Университет им. Ф.М. Достоевского, Омск, Россия

Аннотация

Предлагается схема разделения секрета, использующая представление данных в виде *n*-мерного объекта и переход к его метрическим характеристикам. Рассматривается возможность применения и оптимизация данного алгоритма для защищенной передачи видеоданных. Описывается создание системы фильтров обработки видеоданных для защищенной передачи по сети Internet.

Введение

На сегодняшний день защищенная передача видеоинформации по вычислительной сети является востребованной задачей. Необходимость в такой защите возникает при передаче конфиденциальной информации, организации видеоконференций, организации платной трансляции видео. Для осуществления защищенной передачи необходимо создать механизм, учитывающий большой объем защищаемых данных (видеоролики высокого качества), однородность передаваемых данных и необходимость потоковой обработки данных в реальном времени.

Одним из путей решения этой проблемы является использование криптографических методов защиты, но у них есть ряд недостатков при передаче видеоданных. Во-первых, программная реализация хороших криптографических алгоритмов, таких, как RSA, AES, DES недостаточно быстро обрабатывает большие объемы данных, что делает невозможным обработку потока видео в реальном времени. Во-вторых, однородный характер передаваемых данных (к примеру, соседние кадры видео) дает почву для применения злоумышленником различных методов криптографического анализа.

Другим, более подходящим подходом, является применение схемы разделения секрета [3] для передачи видео. В отличие от криптографических методов, здесь не требуется проведение трудоемких вычислений и применения раундов шифрования. Кроме того, при передаче информации по нескольким маршрутам вычислительной сети с динамической маршрутизацией будет минимизирован риск перехвата злоумышленником достаточного количества данных для восстановления исходного видео [1].

Предлагается построение механизма защиты передачи видеоданных, основывающегося на схеме разделении секрета. Характерной особенностью схемы является использование геометрического объекта, формируемого на основе RGB-представления видеоданных.

Переход к геометрическому описанию данных

Для реализации схемы потребуется переход к геометрическому объекту, содержащему секрет. Для этого будет осуществляться побайтовое преобразование блока обрабатываемых данных F к точкам в n-мерном пространстве [2]. Для этого потребуется, чтобы общий размер защищаемого блока информации в байтах M был кратен n. Если это не выполняется, то необходимо осуществить выравнивание по n байтам с помощью стандартной процедуры выравнивания, подобно тому, как это приводится в алгоритме вычисления хэш-кода MD5.

Будем считать, что M кратно n.

Преобразуем секрет $F = \overline{f_1 f_2 \dots f_M}$, где f_i - байты F, следующим образом: разбиваем F на блоки по n байт, каждый такой блок будет являться точкой в n-мерном пространстве:

$$\begin{cases} A_1 = (f_1, f_2, \dots, f_n) \\ A_2 = (f_{n+1}, f_{n+2}, \dots, f_{2\cdot n}) \\ \dots \\ A_k = (f_{(k-1)\cdot n+1}, f_{(k-1)\cdot n+2}, \dots, f_{k\cdot n}) \\ \dots \ . \end{cases}$$

Полученные точки A_i есть вершины искомого геометрического объекта.

Построение частей секрета по рассматриваемой схеме будет производиться на основе метрических характеристик исходного геометрического объекта. Для описания формата получаемых частей секрета понадобится понятие используемой метрики. В нашем случае это будет евклидово расстояние в n-мерном пространстве, которое для двух точек $A_i = (a_{i1}, \dots, a_{in})$, $A_j = (a_{j1}, \dots, a_{jn})$ вычисляется как

$$\rho(A_i, A_j) = \sqrt{\sum_{k=1}^{n} (a_{ik} - a_{jk})^2} \qquad (1)$$

Исходя из этого, частями секрета для данной схемы будут являться наборы значений метрик $\rho(A_i, A_j)$, которые можно представить в виде непрерывного массива байт.

Следует отметить, что с точки зрения машинных вычислений, для данной схемы более удачными будут не сами евклидовы метрики, а их квадраты. Это уменьшит трудоемкость схемы, т.к. избавит от тру-

доемкой операции извлечения квадратного корня. К тому же, евклидовы метрики, в общем случае, являются иррациональными числами с плавающей точкой, которые требуют больше памяти для хранения, нежели целочисленные данные. Так как координатами исходного объекта являются числа байтового типа, занимающие в памяти 8 бит, то квадрат разности двух координат будет умещаться в 16 бит, а сумма n таких квадратов – в $16+\lceil \log_2 n \rceil$ бит, т.е. не более 3 байт для $n < 256$. Для хранения же самих метрик нужно использовать типы данных с плавающей точкой, для хранения которых необходимо не менее 4 байт.

Исходя из приведенных выше рассуждений, можно оценить, насколько увеличится количество информации при переходе от вершинного описания геометрического объекта к списку метрик. Ниже будет показано, что для такого перехода понадобится n метрик для каждой вершины, которая в свою очередь определяет n целочисленных координат. Таким образом, количество метрик, необходимых для описания геометрического объекта в n-мерном пространстве, совпадает с количеством координат объекта, т.е. с размером секрета в байтах. Следовательно, отношение объема информации, необходимого для хранения метрик, к размеру секрета определяется количеством байт для хранения метрики. То есть если мы выберем евклидово расстояние в качестве метрики, для хранения которого в машинном виде потребуется 4 байта, то общий объем частей секрета будет примерно в 4 раза больше размера исходного секрета, а в случае использования квадратов евклидовых расстояний – в 2-3 раза больше в зависимости от n.

Алгоритм разделения секрета

Ниже представлен алгоритм разделения секрета на n каналов, который генерирует части секрета путем представления данных из разделяемого потока в виде координат точек в n-мерном пространстве с последующим переходом к евклидовым метрикам для полученного объекта. В качестве секрета для каждой последующей вершины используются евклидовы расстояния от нее до n предыдущих различных вершин объекта. Такой переход к метрикам позволит однозначно восстановить все вершины объекта итеративно по уже восстановленным предыдущим вершинам, так как точка в n-мерном пространстве однозначно (с точностью до знака) определяется другими n точками и евклидовыми расстояниями от каждой из этих точек до нее. Иначе говоря, n опорными точками и n жесткими связывающими опорами мы можем жестко зафиксировать точку в n-мерном пространстве.

Алгоритм является итерационным, для каждой итерации можно определить последовательность шагов.

Вход: $A_1, A_2, \ldots, A_l$ - список вершин геометрического объекта в n-мерном пространстве, $l = M / n$ - количество вершин.

Выход: $S_1, S_2, \ldots, S_n$ - сформированные части секрета, состоящие из значений евклидовых метрик исходного объекта. K - «ключевая» часть секрета, здесь хранятся инициализирующие данные для алгоритмов формирования и восстановления секрета. В протоколе передачи данных K может быть как открытой информацией (даже при ее наличии атака злоумышленника на алгоритм останется вычислительно сложной задачей перебора), так и отдельной частью секрета, отличающейся от остальных частей размером (без информации об инициализирующих алгоритм переменных восстановление не сможет быть однозначным).

Инициализация. На этом шаге формируются значения «стартовых» точек $A_{-n+1}, A_{-n+2}, \ldots, A_0$. Эти точки понадобятся для перехода к метрикам для n первых вершин объекта. Это могут быть значения, сгенерированные с помощью криптостойкого генератора псевдослучайных чисел. Сгенерированные значения помещаются в ключевой массив K.

$i := 0$ - инициализируем номер обрабатываемой на данный момент вершины объекта.

$S_1, S_2, \ldots, S_n$ - массивы для частей секрета, делаем их пустыми.

Шаг 1. Переход на новую итерацию.

$i := i + 1$. Выбираем для обработки вершину A_i и предшествующие n вершин $A_{i-n}, A_{i-n+1}, \ldots, A_{i-1}$.

Шаг 2. Обрабатываем вершину A_i. Для нее вычисляются величины $R_{i1}, \ldots R_{in}$, являющиеся квадратами расстояний от A_i до каждой из предыдущих n точек:

$$\begin{cases} R_{i1} = (\rho(A_{i-n}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-n-1)\cdot n+j})^2 \\ R_{i2} = (\rho(A_{i-n+1}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-n)\cdot n+j})^2 \\ \ldots \\ R_{in} = (\rho(A_{i-1}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-2)\cdot n+j})^2 \end{cases} \quad (2)$$

Как было показано выше, здесь более целесообразно использовать квадраты евклидовых расстояний, так как это уменьшает трудоемкость вычислений и размер частей секрета.

Полученные метрики $R_{i1}, \ldots R_{in}$ позволяют восстановить точку в n-мерном пространстве с точностью до симметрии, т.к. при возведении расстояний в квадрат знак самого расстояния по координатам пропадает. Для этого на шаге 3 мы будем формировать значения, позволяющие восстановить однозначное расположение точки A_i.

Шаг 3. Для вершины A_i и предшествующих ей вершин $A_{i-n}, A_{i-n+1}, \ldots, A_{i-1}$ вычисляем k_i - знак n-

мерного векторного произведения векторов $\overrightarrow{A_{i-j}A_i}$, который помещается в массив K. В зависимости от значения k_i при восстановлении будет выбираться одно из двух решений системы (2).

$$k_i = sign(\overrightarrow{A_{i-n}A_i} \otimes \overrightarrow{A_{i-n+1}A_i} \otimes \ldots \otimes \overrightarrow{A_{i-1}A_i}) \quad (3)$$

Шаг 4. Формирование частей секрета. Дописываем полученные на предыдущем шаге расстояния R_{ij} к частям секрета $S_1, S_2, \ldots, S_n$ по принципу $R_{ij} \to S_j$.

Шаг 5. Если в списке вершин объекта еще остались необработанные вершины, возвращаемся к шагу 1.

Алгоритм восстановления секрета

В то время как описанный выше алгоритм разделения является достаточно быстрым и генерация частей секрета является легко решаемой задачей, представленный ниже алгоритм восстановления работает несколько медленнее, т.к. в общем случае восстановление сводится к решению системы уравнений 2-й степени (2). Лишь в случае $n = 2$ и $n = 3$ задачу можно свести к геометрическому алгоритму, находящему точное решение за достаточно малое время.

На вход алгоритма восстановления поступают n частей секрета и «ключевой» массив K, содержащий значения координат n стартовых точек и значения параметра k_i для выбора из двух возможных вариантов получающихся в результате решения системы (2) точек.

Подобно алгоритму разделения секрета, алгоритм восстановления является итерационным, на каждом последующем шаге будет находиться одна из точек геометрического объекта, являющегося секретом.

Вход: $S_1, S_2, \ldots, S_n$ - сформированные части секрета, состоящие из значений евклидовых метрик исходного объекта. K - «ключевая» часть секрета, здесь хранятся инициализирующие данные для алгоритмов формирования и восстановления секрета.

Выход: $A_1, A_2, \ldots, A_l$ - список вершин исходного геометрического объекта в n-мерном пространстве, $l = \frac{M}{n}$ - количество вершин, зная длину l, легко находим размер исходного секрета в байтах.

Шаг 0. Инициализация. На этом шаге в память заносятся значения координат «стартовых» точек $A_{-n+1}, A_{-n+2}, \ldots, A_0$. Также инициализируется счетчик по i.

Шаг 1. Переход на следующую итерацию.

$i := i + 1$; Текущей вершиной становится вершина A_i. Для обработки выбираем предшествующие ей n вершин $A_{i-n}, A_{i-n+1}, \ldots, A_{i-1}$. Т.к. алгоритм проходит вершины последовательно, все предшествующие A_i вершины уже восстановлены, либо изъяты из ключевого массива K.

Шаг 2. Получение частей секрета для восстановления текущей вершины.

Заносим в переменные R_{ij} значения из частей секрета S_j, $j = 1 \ldots n$. После этого смещаем указатели в частях секрета S_j на одну позицию, т.е. так, чтобы они указывали на следующие элементы частей секрета. В k_i заносим очередное значение из массива K.

Шаг 3. Восстановление секрета.

Для нахождения координат точки A_i путем решения системы (2) вычисляем значения координат двух точек $A_i^{'}, A_i^{''}$, удовлетворяющих (2). Для выбора искомой вершины A_i вычисляем

$$k_i' = sign((\overrightarrow{A_{i-n}A_i}) \otimes (\overrightarrow{A_{i-n+1}A_i}) \otimes \ldots \otimes (\overrightarrow{A_{i-1}A_i})) \quad (4)$$

$$A_i = \begin{cases} A_i', \textit{если } k_i' = k_i \\ A_i'', \textit{иначе} \end{cases} \quad (5)$$

Для решения системы уравнений (2) используется метод Ньютона, который не дает точного решения, но достаточно быстро находит приближенное, а поскольку координаты точек являются целочисленными, то с помощью метода Ньютона можно быстро определить, к какому целочисленному значению оно стремится.

Шаг 4. Если в списке вершин объекта еще остались необработанные вершины, возвращаемся к шагу 1.

После восстановления вершинного описания исходного геометрического объекта достаточно просто осуществляется переход к байтовой записи секрета.

$$\begin{cases} A_1 = (f_1, f_2, \ldots, f_n) \\ A_2 = (f_{n+1}, f_{n+2}, \ldots, f_{2 \cdot n}) \\ \ldots \\ A_l = (f_{(l-1) \cdot n+1}, f_{(l-1) \cdot n+2}, \ldots, f_{l \cdot n}) \end{cases} \quad (6)$$

Искомый секрет - $F = \overline{f_1 f_2 \ldots f_M}$, $M = ln$.

Отметим, что на шаге 3 приведенного алгоритма однозначно восстановить исходную точку по имеющемуся набору из n предыдущих точек и расстояний R_{ij} не всегда возможно. В случае, когда все n

предыдущих точек лежат в одной гиперплоскости размерности $n-2$, точки, удовлетворяющие (2), будут определяться с точностью до поворота на произвольный угол относительно указанной гиперплоскости. В этом случае необходимо использовать дополнительную метрику (например, угол поворота φ относительно данной гиперплоскости), и указывать ее вместо одного из расстояний R_{ij}, где *j*-я предыдущая точка является линейной комбинацией остальных точек, и удаление этой точки из рассмотрения на данной итерации не повлияет на размерность указанной гиперплоскости.

Анализ защищенности схемы

Ниже будут рассмотрены атаки на схему, направленные на восстановление сообщения по одной или нескольким частям секрета. Рассматриваются различные возможности злоумышленника по получению дополнительной информации о секрете, и методы улучшения схемы, направленные на противостояние различным атакам.

Один из типов атаки, который может быть применен к описанной схеме, основывается на следующей особенности алгоритма: при разделении секрета для вычисления частей R_{ij} используются точные целые значения расстояний между вершинами по каждой из осей координат. Таким образом, для каждой R_{ij} известно, что она является суммой квадратов *n* целых чисел, и для не очень больших значений *n* злоумышленник может найти набор точных смещений очередной точки относительно *i*-й предыдущей, что облегчает восстановление секрета в целом. Для устранения этой проблемы можно при вычислении R_{ij} фигурирующие в (2) расстояния учитывать со случайной погрешностью $\varepsilon_{ij} \in (-0,5 \ldots 0,5)$

$$\begin{cases} R_{i1} = (\tilde{\rho}(A_{i-n}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-n-1)\cdot n+j} + \varepsilon_{ij})^2 \\ R_{i2} = (\tilde{\rho}(A_{i-n+1}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-n)\cdot n+j} + \varepsilon_{ij})^2 \\ \ldots \\ R_{in} = (\tilde{\rho}(A_{i-1}, A_i))^2 = \sum\limits_{j=1}^{n} (f_{(i-1)\cdot n+j} - f_{(i-2)\cdot n+j} + \varepsilon_{ij})^2 \end{cases} \quad (7)$$

В этом случае величины R_{ij}, составляющие части секрета, будут приближенными значениями суммы квадратов, и для восстановления отдельных расстояний злоумышленнику придется перебирать все варианты на некоторой *n*-мерной сфере.

Еще одна уязвимость представленной схемы появляется из-за «слабых» участков секрета. Они возникают в случае, когда среди $n+1$ точек, использующихся для формирования очередных значений R_{ij}, попадаются одинаковые. В частности, текущая обрабатываемая точка A_i может совпадать с одной из предыдущих *n* точек, что автоматически обнуляет значение метрики, идущее в одну из частей секрета, и злоумышленник, получив эту часть, может выявить совпадения точек, и исходя из этого получить закономерности для восстанавливаемого секрета. Эту уязвимость можно ликвидировать, модифицировав в алгоритмах разделения и восстановления секрета условие выбора *n* предшествующих точек следующим образом: необходимо выбирать *n* ближайших предшествующих различных точек, при условии, что все стартовые точки различны. Описанная уязвимость является существенной для передачи реального видео, т.к. в этом случае наличие однородных областей в изображении автоматически будет давать слабые участки.

Применение схемы для передачи видеоданных

Так как необходима передача кадров видео на лету, то на вход алгоритма разделения они будут поступать в несжатом виде, т.е. в формате RGB-массива точек (пикселей), расположенного в памяти линейно. Для 24-битной цветовой схемы получаем по 1 байту на каждую составляющую цвета. В таком виде изображение будет поступать на вход схемы разделения секрета с $n=3$, каждый пиксель изображения будет соответствовать вершине трехмерного объекта, ее координатами будут, соответственно, красная, зеленая и синяя составляющие цвета. После отработки алгоритма разделения секрета, полученные части можно вновь интерпретировать как изображения того же формата (RGB24), используя метрики в качестве цветов пикселей. После передачи сформированных кадров по сети, они поступают на вход алгоритма восстановления секрета, на выходе которого мы получим исходное изображение. Таким образом, рассматриваемая схема разделения секрета при $n=3$ легко приспосабливается для прямого преобразования видео в видео, т.е. можно создать фильтры преобразования видео, реализовывающие алгоритмы разделения и восстановления секрета. На их основе формируется рабочее приложение, запускающее граф фильтров на основе DirectShow.

У такого способа безопасной передачи видеоинформации есть достаточно серьезная уязвимость, о которой было сказано раньше. Изображения идут на вход алгоритма разделения секрета в несжатом виде. Кроме того, реальное изображение часто содержит большие области одного цвета (или плавно переходящего оттенка цвета), таким образом, при представлении изображения в виде геометрического объекта мы получим большое количество вершин с одинаковыми либо соседними координатами, что сразу дает «слабые места» для алгоритма разделения секрета. Во избежание этого эффекта можно поставить перед фильтром разделением секрета фильтр видеосжатия (например, MPEG-4), но такая модификация, во-первых, повлияет на скорость передачи

видео, так как видеосжатие без потерь является достаточно трудоемкой задачей, во-вторых, это не уберет, в полной мере, корреляцию между цветами соседних пикселей.

Предлагается другое, более оптимальное и быстрое решение: использовать в качестве исходного изображения для секрета кадр, пиксели которого берутся поочередно из кадров нескольких различных потоков видео. В этом кадре любые 2 соседних пикселя взяты из разных изображений, их цветовые компоненты не связаны между собой, следовательно, при разделении секрета не будет больших «слабых» участков, дающих дополнительную информацию злоумышленнику.

После восстановления такого кадра видео, каждый из его прообразов легко восстановить с помощью интерполяции недостающих пикселей по известным. Потеря качества, возникающая при этом, гораздо ниже, чем при использовании алгоритмов сжатия видео, т.к. здесь хранится большая доля пикселей каждого изображения. Обратим внимание, что смешивание потоков видео приводит к «голографичности» суммарного потока и согласно предложенной методике возможна одновременная передача нескольких потоков в одном, причем практически без потерь информации.

Набор фильтров для потоковой обработки видео реализован на базе технологии Microsoft DirectShow с помощью программной среды Microsoft Visual Studio. Набор фильтров включает в себя фильтр для объединения нескольких потоков видео в один, фильтр, осуществляющий обратное преобразование, и фильтры, реализующие разделение секрета и восстановление секрета. Для передачи частей секрета используются фильтры Microsoft DirectShow ASF Writer и ASF Reader.

Литература

1. Ефимов В.И., Файзуллин Р.Т. Система мультиплексирования разнесенного TCP/IP трафика // Вестник ТГУ. Приложение. – 2005. -№ 14. -С. 115-117.
2. Свенч А.А., Файзуллин Р.Т. Представление геометрического объекта списком метрических характеристик // Математические методы распознавания образов: 12-я Всероссийская конференция: Сборник докладов. М.: МАКС Пресс, 2005. –С. 434-437.
3. Shamir A. How to share a secret // Communications of the ACM 22, 1979. С. 612-613.

SECRET SHARING SCHEME BASED ON THE METRIC CHARACTERISTICS OF DATA FOR SECURE TRANSMISSION OF VIDEO CONFERENCING

A.A. Svench[1], R.T. Faizullin[1]

[1]Omsk State University (OmSU), Omsk, Russia

Abstract:

We propose a secret sharing scheme based on presenting data in the form of an n-dimensional object and on changing the metric characteristics. We look into opportunities of application and optimization of this algorithm for secure transmission of video conferencing. We describe a system of video data processing filters for secure transmission over the Internet.

Keywords: secure transmission, video conferencing, data processing, filters, metric characteristics.

Citation: Svench AA, Faizullin RT. Secret sharing scheme based on the metric characteristics of data for secure transmission of video conferencing [In Russian]. Computer Optics 2007; 31(1): 47-51.

References:

[1] Efimov VI, Faizullin RT. A multiplexed system for TCP/IP traffic [In Russia]. Supplement to Tomsk State University Journal Supplement 2005; 14: 115-117.

[2] Svetch AA, Faizullin RT. Representation of a geometric object for a list of metric characteristics [In Russian]. Proceedings of the 12th All-Russia Conference “Mathematical Methods for Pattern Recognition.” Moscow: MAX Press Publisher 2005: 434-437.

[3] Shamir A. How to share a secret // Communications of the ACM 22, 1979. С. 612-613.