Лысенков С.А.

РАЗРАБОТКА И АНАЛИЗ АЛГОРИТМА РЕЧЕВОГО КОДИРОВАНИЯ С СИСТЕМОЙ ЗАЩИТЫ ИНФОРМАЦИИ ДЛЯ СЕТЕЙ VoIP НА ОСНОВЕ ТЕХНОЛОГИИ ОДКС

Сеть VoIP является критичной к времени задержки. Задержка в сети VoIP рассчитывается по следующей формуле:

$$Z_{об} = Z_{код} + Z_{инк} + Z_{оч} + Z_{пер} + Z_{сети} + Z_{пер} + Z_{рас} + Z_{дек},$$

где Z задержки вносимые: $Z_{код}$ – алгоритмом кодирования, $Z_{инк}$ – инкапсуляцией пакетов, $Z_{оч}$ – постановкой в очередь, $Z_{пер}$ – передачей, $Z_{сети}$ – сетью, $Z_{рас}$ – распаковкой пакета, $Z_{дек}$ – алгоритмом декодирования. Суммарная величина задержки не должна превышать 250 мс, то есть $Z_{об} \le 250$мс.

При включение в IP сеть средств защиты возникает дополнительная задержка $Z_{доп.}$

$$Z_{об} = Z_{код} + Z_{инк} + Z_{оч} + Z_{пер} + Z_{сети} + Z_{пер} + Z_{рас} + Z_{дек} + Z_{доп}$$

Рассмотрим возможность расположения средств защиты информации в сети VoIP на следующих этапах: до речевого шлюза; после речевого шлюза; непосредственно в речевом шлюзе.

При расположении средств защиты информации в сети VoIP до речевого шлюза возникают следующие задержки: на передающей стороне - время работы алгоритма шифрования; на принимающей - время работы алгоритма дешифрования.

$$Z_{доп} = Z_{шиф} + Z_{дешиф}, \quad (1)$$

где Z задержки вносимые: $Z_{шиф}$ – алгоритмом шифрования, $Z_{дешиф}$ – алгоритмом дешифрования.

При расположении средств защиты информации после речевого кодека, до упаковки в IP пакеты, то есть в речевом шлюзе возникают следующие задержки: на передающей стороне - время работы алгоритма шифрования; на принимающей - время работы алгоритма дешифрования (1).

Возможно расположение средств защиты информации - после речевого шлюза. В этом случае на вход устройства защиты информации речевой сигнал будет поступать в упакованном виде (IP пакет). Далее возможны два варианта:

- в первом случае происходит распаковка IP пакета, его шифрование и инкапсуляция зашифрованного сообщения. При этом возникают следующие задержки на передающей стороне: распаковка IP пакета; время работы алгоритма шифрования; инкапсуляция сообщения в IP пакет. На принимающей стороне: распаковка IP пакета; время работы алгоритма дешифрования; инкапсуляция сообщения в IP пакет.
$$Z_{доп} = Z_{рас} + Z_{шиф} + Z_{инк} + Z_{рас} + Z_{дешиф} + Z_{инк}; \quad (2)$$
- во втором случае происходит шифрование всего IP пакета и инкапсуляция зашифрованного сообщения в пакет. При этом задержки распределены следующим образом на передающей стороне: время работы алгоритма шифрования, инкапсуляция сообщения в пакет. На принимающей стороне: распаковка IP пакета и время работы алгоритма дешифрования. Данный способ организации защиты увеличивает объем служебной информации в 2 раза.
$$Z_{доп} = Z_{шиф} + Z_{инк} + Z_{рас} + Z_{дешиф} \quad (3)$$

Сравнивая выражения (1), (2), (3) получаем, что величина дополнительной задержки будет меньше в (1), то есть при расположении средств защиты до речевого шлюза или непосредственно в речевом шлюзе. В первом случае в качестве средств защиты возможно использовать аналоговые скремблеры, но они вносят большую задержку в канал связи. Например, в скремблерах с временной инверсией задержка составляет около 500 мс при допустимой $Z_{об} \le 250$ мс. Главная проблема заключается в том, что на приемной стороне в случае потери передаваемой кодером битовой посылки,

исходные данные для речевого синтезатора получаются интерполяцией данных с предыдущих “хороших” кадров, а так как аналоговые скремблеры преобразуют исходный речевой сигнал посредством изменения его амплитудных, частотных и временных параметров в различных комбинациях, то восстановленная речь не будет обладать достаточной разборчивостью. При определенных условиях возможна полная потеря качества речи и абсолютная неразборчивость.

Во втором случае, при реализации средств защиты информации после речевого кодека, до упаковки в IP пакеты, возможно применение различных средств криптографической защиты. Такой вариант имеет следующие преимущества: защита канала «точка - точка»; универсальность, не важно какой протокол используется для передачи данных, как следствие отсутствие проблем согласования; минимизация времени задержки вносимой в канал связи (время дополнительной задержки вносимой в канал связи – это только время работы алгоритма).

Для расположения средств защиты информации в сети VoIP наилучшим является этап речевого кодирования.

Существует три основных класса криптографических алгоритмов защиты информации: потоковые, блочные и шифры с открытым ключом. Блочные алгоритмы шифруют информацию блоками, следовательно будет вноситься дополнительная задержка на формирование блока. Алгоритмы с открытым ключом требуют большой вычислительной мощности и вносят огромную дополнительную задержку из за сложности алгоритма. Потоковые шифры шифруют информацию по битно и вносят минимальную задержку. Следовательно, они являются наиболее подходящими для использования в сети VoIP. Среди потоковых шифров выбран алгоритм SEAL. Особенностью SEAL является то, что он представляет собой семейство псевдослучайных функций. SEAL удобен в приложениях, где неприменимы традиционные потоковые шифры. Его отличие состоит в том, что можно легко получить доступ к любой позиции потока ключей. Семейство псевдослучайных функций также упрощает проблему синхронизации, на принимающей стороне не нужно хранить состояние шифра.

Из кодеков, стандартизированных ITU – T, наиболее предпочтительным с точки зрения соотношения качество речи/скорость потока является алгоритм G.723.1 (ACELP).

При создании алгоритма учитывалось, что при поступлении речевого сигнала на вход вокодера, осуществляется буферизация данных. Вокодер обрабатывает речевой сигнал кадрами длиной минимум 10мс. Следовательно, некоторое время при поступлении сигнала система простаивает, она ожидает накопления данных, необходимых для начала работы. Но это время простоя можно эффективно использовать для выполнения криптографическим алгоритмом предварительных действий с ключом, без ущерба производительности вокодера. Как следствие мы избегаем медленных операций таких, как функция сжатия алгоритма SHA, вносящих дополнительную значительную задержку в канал связи. Схема алгоритма речевого кодирования со встроенной системой защиты информации представлена на рисунке 1.

В вокодере применяется векторное квантование параметров линейного предсказания по кодовой книге размерностью 1024 центроида. В существующих вокодерах поиск центроидов в кодовой книге осуществляется с помощью алгоритма бинарного поиска. Длина внешнего пути бинарного дерева D_n, соответствующего алгоритму бинарного поиска, равна:

$$D_n = (N+1)(\log_2 N + 2) - 2^{\log_2 N + 1},$$

где N – количество записей в кодовой книге.

Учитывая, что все аргументы поиска равновероятны можно определить среднее число сравнений C_N. В общем случае, если $k=log_2N$ имеем

$$C_N = k + 1 - \frac{(2^{k+1} - k - 2)}{N}.$$

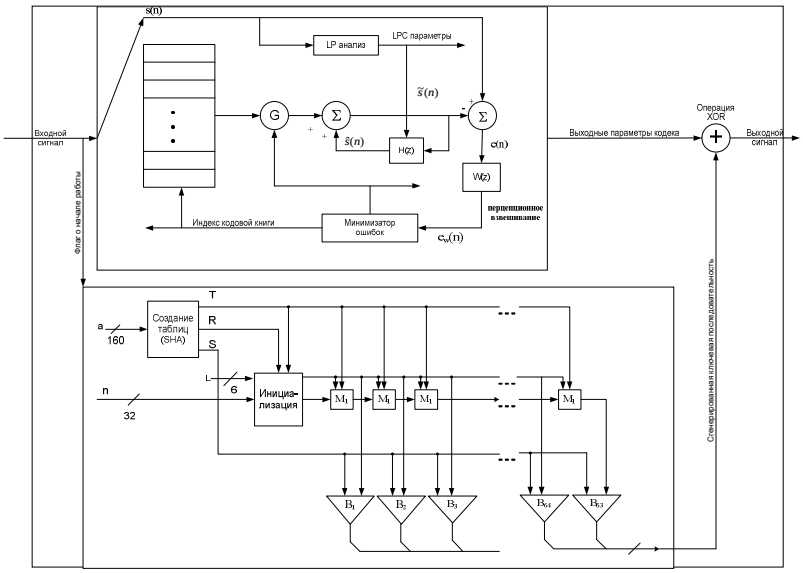

Рисунок 1. Схема работы алгоритма речевого кодирования со встроенной системой защиты на передающей стороне.

Алгоритм бинарного поиска требует максимум (log_2N+1) сравнений, среднее число сравнений при удачном поиске приближено равно (log_2N-1). Среднее время работы программы бинарного поиска T_{p1} составляет:

$$T_{p1} = (18 \cdot \log_2 N - 15)u, \quad (4)$$

где u – нижняя граница поиска по бинарному дереву.

В качестве модификации предлагается применять модифицированный алгоритм поиска со вставкой по дереву. В отличие от бинарного поиска предполагается, что узлы дерева содержат следующие поля, позволяющие уменьшить время поиска:

- ключ, хранящийся в узле;
- указатель на левое поддерево узла;
- указатель на правое поддерево узла.

Время поиска T_{p2} равно

$$T_{p2} = (6.5C - 2.5S + 5)u, \quad (5)$$

где C- число произведенных сравнений, S=1 при удачном поиске.

Сравним (4) и (5), то есть время поиска центроидов в кодовой книге по стандартному и модернизированному алгоритму поиска, критерий K:

$$K = \frac{T_{p1}}{T_{p2}} = \frac{(18 \cdot \log_2 N - 15)\text{u}}{(6.5\text{C} - 2.5\text{S} + 5)\text{u}} = 2.44$$

При использовании предложенного алгоритма время выполнения поиска уменьшилось в 2.44 раза.

Увеличить быстродействие алгоритма возможно за счет технологии ОДКС. Параллельное выполнение нескольких команд позволит увеличить скорость работы алгоритма, а как следствие уменьшить величину задержки вносимой в канал связи. Некоторые команды нельзя выполнять одновременно с другими, связано это с тем, что эти команды используют результаты предыдущих операций и возвращаемый ими результат используется в следующих инструкциях.

Наиболее медленной операцией криптографического алгоритма Seal является алгоритм генерации ключевой последовательности. На рисунке 2 показана схема возможного параллельного выполнения команд. Блоки стоящие рядом на одном уровне означают параллельно выполняющиеся команды. Некоторые команды нельзя выполнять одновременно с другими, связано это с тем, что эти команды используют результаты

предыдущих операций и возвращаемый ими результат используется в следующих инструкциях.

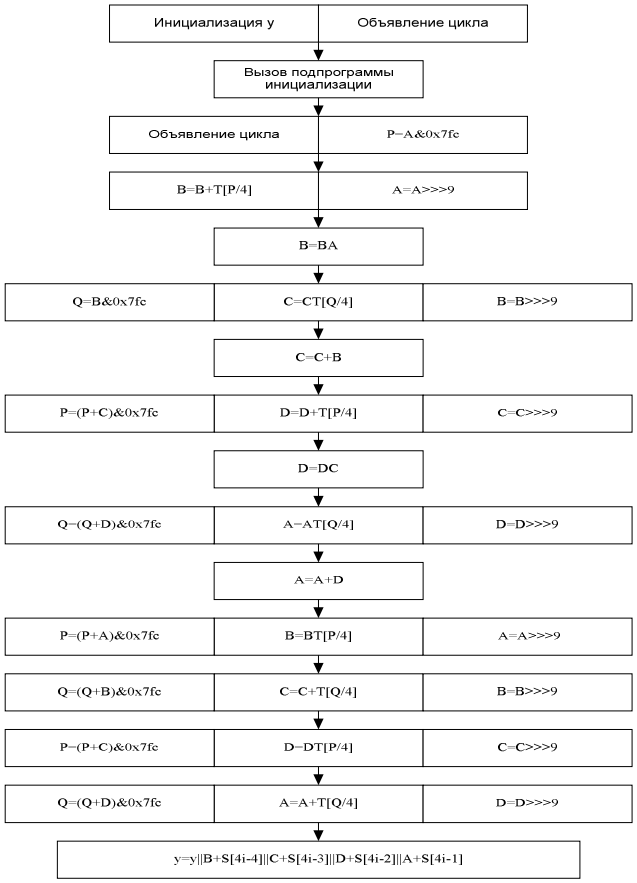

Рисунок 2 Параллельное выполнение алгоритма SEAL.

При параллельном выполнении команд выигрыш в скорости не составляет, как можно подумать, несколько раз. При схеме изображенной на рисунке 3.10 уменьшение времени выполнения программы в реальных условиях составит 30-60%. Предсказать более точный процент сокращения времени работы программы невозможно, по некоторым причинам: нельзя предвидеть состояние памяти до выполнения нашей программы, регистрового файла, состояния системных прерываний, занятость некоторых АЛУ и т.д.Например, если выполнять три команды сложения различных операндов, то время выполнения составит 482 циклов ЦПУ (около 9,6*10-7 секунд при использовании процессора TMS320C64x), а при их выполнении параллельно потребуется 283 цикла ЦПУ (около 5,7*10-7 секунд), то есть время исполнения программы сокращается в 1,7 раза. Для достижения быстродействия необходимо также распараллелить выполнение процедуры инициализации алгоритма SEAL.

Криптографический алгоритм SEAL состоит из 33 команд. Блок инициализации криптографического алгоритма содержит 30 команд. После распараллеливания общее количество команд составило 28. Таким образом процент распараллеливания, показывающий на сколько уменьшилось количество команд, составил:

$$V = \frac{m_1 - m_2}{m_1} \cdot 100\% \approx 56\%,$$

где V – показатель, определяющий процент распараллеливания, m_1 – количество команд в последовательной программе, m_2 – количество команд в параллельной программе.

Компьютерное моделирование осуществлялось в интегрированной среде разработчика Code Composer Studio 3.1 Platinum Edition. Для проведения сравнительного анализа осуществлено моделирование алгоритма речевого кодирования G.723.1 (ACELP),

кодека со встроенной системой защиты информации выполняющегося последовательно и выполняющегося параллельно.

В результате экспериментальных исследований установлено, что минимальное время задержки в канал связи вносится при реализации системы защиты информации и кодеков на TMS320C64xx. Программная модель речевого кодека со встроенной системой защиты информации функционирует в режиме реального времени. Полученные экспериментальным путем данные представлены на рисунке 3.

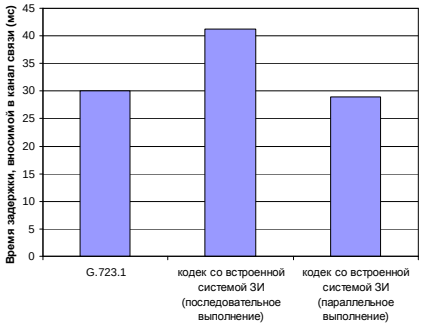

Рисунок 3. Время задержки, вносимой в канал связи, программными моделями кодеков.

Экспериментальные данные подтверждают, что разработанная программная модель речевого кодека со встроенной системой защиты информации вносит меньшую задержку, чем кодек G.723.1, стандартизированный ITU-T.

Полученные данные позволяют оценить увеличение быстродействия системы при параллельном выполнении алгоритма.

$$R = \frac{t_1 - t_2}{t_1} \cdot 100\% \approx 30\%,$$

где t_1 – время работы кодека со встроенной системой защиты информации при последовательном выполнении, t_2 – время работы кодека со встроенной системой защиты информации при параллельном выполнении, R – критерий показывающий на сколько процентов уменьшилось время выполнения алгоритма.

Таким образом, разработанный алгоритм позволяет создать систему защиты информации от несанкционированного доступа в сетях VoIP, без внесения дополнительной задержки в канал связи.