перекрывались либо наполовину, либо на 2/3. Перекрытие используется для предотвращения потери информации о сигнале на границе.

Для вычисления мел-частотных кестральных коэффициентов, на вход алгоритма подаётся последовательность отсчётов участка сигнала, исследуемого на данной итерации. К данной последовательности применяется весовая функция и затем дискретное преобразование Фурье. Весовая функция используется для уменьшения искажений в Фурье анализе, вызванных конечностью выборки. В качестве весовой функции используется окно Хэммига.

Полученное представление сигнала в частотной области разбивают на диапазоны с помощью банка треугольных фильтров. Границы фильтров рассчитывают в шкале мел. Данная шкала является результатом исследований по способности человеческого уха к восприятию звуков на различных частотах. Перевод в мел-частотную область осуществляется по формуле $B(f)=1127\ln(1+f/700)$.

Количество мел-частотных кепстральных коэффициентов определяется количеством треугольных фильтров. Фильтры применяются к квадратам модулей коэффициентов преобразования Фурье. Полученные значения логарифмируются. Заключительным этапом в вычислении мел-частотных кепстральных коэффициентов является дискретное косинусное преобразование.

ТЕСТИРОВАНИЕ НА ПРОСТОТУ БОЛЬШИХ ЧИСЕЛ СПЕЦИАЛЬНОГО ВИДА

А.В. ИВАШКЕВИЧ, Е.Д. СТРОЙНИКОВА

С целью создания генератора псевдопростых чисел были реализованы следующие тесты проверки чисел на простоту: Ферма, Миллера–Рабина, Соловея–Штрассена, Лукаса, BPSW. Первые три указанных теста являются вероятностными. Они позволяют очень эффективно отбраковать составные числа, однако не в состоянии строго доказать простоту числа, а лишь позволяют говорить, что число p не является составным с некоторой вероятностью. Наиболее эффективным из этих трех алгоритмов является тест Миллера–Рабина.

Верхняя граница ошибки на одной итерации для теста Миллера–Рабина в 2 раза меньше аналогичной для теста Соловея–Штрассена и в 4 раза — верхней границы ошибки для теста Ферма. Если на одной итерации вероятность ошибочного решения в тесте не превышает 1/4, то на двух итерациях — 1/16, на трех — 1/64. для того чтобы вероятность ошибки не превышала 0,0001, требуется всего 7 итераций, что в 2 раза меньше, чем для теста Соловея–Штрассена.

На основании вышеуказанных алгоритмов был разработан программный модуль генерации простых чисел заданной длины. Для его создания была использованна среда Microsoft Visual Studio 2010 и язык программирования C#.

Основной сложностью при создании модуля генерации стала реализация алгоритмов проверки чисел на простоту.

В результате выполенной работы были получены следующие средние временные результаты генерирования псевдопростых чисел: число длиной 256 бит было сгенерированно за 1 секунду, 512 бит — за 2–3 секунды, 1024 бит — за 55 секунд, 2048 бит — за 600 секунд, 3076 бит — за 7200 секунд.

Созданный программный модуль имеет очень важное практическое применение, так как простые числа являются неотъемлемой частью криптографических алгоритмов, используемых для защиты информации.