РАЗРАБОТКА ТЕСТОВЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ДЛЯ ОБНАРУЖЕНИЯ УМЫШЛЕННЫХ ОШИБОК В ПРОГРАММНЫХ РЕАЛИЗАЦИЯХ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ

А.А. Тиунчик

Государственное предприятие "НИИ ТЗИ", Минск, Беларусь

Сложность разработки и тестирования программного обеспечения, реализующего криптографические алгоритмы, в значительной степени обусловлена сложностью самих криптографических алгоритмов. Дополнительным фактором, усложняющим тестирование реализаций таких алгоритмов, является использование разработчиками методов оптимизации программного кода.

Множество ошибок, вносимых разработчиками программного обеспечения при написании программных продуктов, может быть условно разделено на несколько групп:

— случайные ошибки, связанные с логическими ошибками реализации криптографического алгоритма:

— случайные ошибки, вносимые при написании программного кода;

— случайные ошибки, обусловленные неверной организацией ввода и вывода;

— умышленные ошибки, связанные с внесением в программный код недокументированных возможностей;

— умышленные ошибки, связанные с искажением выполнения криптографического алгоритма в целях повышения быстродействия программной реализации.

Нахождение умышленных ошибок является одной из наиболее трудоемких задач тестирования. Для обнаружения таких ошибок часто применяется медленный и дорогостоящий метод анализа исходных текстов. В то же время предварительный анализ реализуемого криптографического алгоритма в ряде случаев позволяет предсказать, какие элементы этого алгоритма представляют наибольшую потенциальную опасность с точки зрения возможности внесения в них умышленных ошибок.

Процесс разработки тестовых последовательностей для обнаружения умышленных ошибок в программных реализациях криптографических алгоритмов может быть разделен на три этапа:

1) нахождение элементов алгоритма, потенциально опасных с точки зрения возможности внесения умышленных ошибок;

2) генерация числовых последовательностей, позволяющих однозначно установить факт существования предполагаемой умышленной ошибки в исследуемом участке программного кода;

3) разработка механизма подачи сгенерированной тестовой числовой последовательности на требуемый участок программного кода, что позволяет тестировать всю реализацию криптографического алгоритма как "черный ящик".

Сгенерированные таким образом последовательности позволяют существенно ускорить и повысить качество тестирования криптографических алгоритмов.