Реализация контроля целостности данных посредством хэш-функции на ПЛИС

Н. С. Савельев, email: nssavelyev01@ya.ru
А. О. Лачинов, email: avatar31221@yandex.ru
Н. А. Сипатров, email: nikita.Sipatrov@yandex.ru
А. Д. Ваничкин, email: vanamaka@mail.ru
Т. В. Стариков, email: TVS_7@mail.ru

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М. Штеменко

Аннотация. *В данной работе рассматривается аппаратная реализация на ПЛИС модуля контроля целостности данных посредством хэш-функции. Поведенческое описание модуля представлено на языке VHDL.*

Ключевые слова: *ПЛИС, хэш-функция, VHDL, Стрибог, ГОСТ Р 34.11-2012.*

Введение

В связи с тем, что осуществление корректного разграничения доступа к информации при современном уровне технологий практически невозможно, становится актуальна проблема выявления несанкционированных действий над информацией. На прикладном уровне решается задача контроля целостности основанная на реализации функций сравнения с эталоном.

Одним из наиболее простых способов осуществления контроля целостности является метод вычисления контрольной суммы данных [1]. В случаях, когда необходима высокая скорость обработки или, когда объём данных не известен, достаточно применение контрольной суммы, однако всегда возможно подобрать такие массивы данных, при которых контрольная сумма будет одинакова. В связи с этим возникает потребность в криптостойких методах. Современные криптографические методы позволяют контролировать целостность хранимых или передаваемых данных посредством хэш-функций [1-5].

Для решения задачи контроля целостности данных, для каждого набора данных вычисляется значение хэш-функции, которое хранится и передается вместе с данными. После чтения данных вычисляется значение свертки и сравнивается с имеющимся контрольным значением. При несовпадении значений можно сделать вывод, что данные были изменены.

Для реализации модуля контроля целостности данных, была выбрана хэш-функция «Стрибог» описанная в [6, 7]. Стандарт [6] определяет алгоритм и процедуру вычисления хэш-функции для любой последовательности двоичных символов, которые применяются в криптографических методах обработки и защиты информации, в том числе для реализации процедур обеспечения целостности, аутентичности, электронной подписи (ЭП) при передаче, обработке и хранении информации в автоматизированных системах [8-11].

Описание и реализация алгоритма преобразования хэш-функции «Стрибог»

Согласно стандарту, реализуемый модуль обрабатывает блоки сообщений длиной 512 бит и вычисляет 512-битные хэш-значения. Последнее ⌊ -битное сообщение дополняется до размера, кратного 512 битам. Присоединяется единичный бит к концу сообщения, после которого следует 512 – 1 – (⌊ mod 512) нулевых бит.

$$m = (m_t) \,||\, (m_{t-1}) \ldots (m_1), \tag{1}$$

где m – сообщение, состоящее из t блоков после дополнения, t – число кратное 512. Схема вычисления хэш-функции H(M) представлена на рисунке 1. Функция сжатия $g_N(h, m)$

$$g_N(h, m) = E(LPS(h \oplus N), m) \oplus h \oplus m \tag{2}$$

где L – умножение справа на матрицу A над полем Галуа GF(2) и есть матрица линейного преобразования A состоящая из 64 восьмибайтовых чисел, P – простая перестановка байтов в исходном массиве в соответствии с правилом, определяемым массивом Tau размером в 64 байта, S – Нелинейное биективное преобразование (при биективном отображении каждому элементу одного множества соответствует ровно один элемент другого множества, то есть это подстановка байтов в исходном векторе по определенному правилу. В данном случае правило задается массивом из 256 значений), E – функция блочного шифра.

$$E(K, m) = X[K_{13}]LPSX[K_{12}]\ldots LPSX[K_2]LPSX[K_1](m) \tag{3}$$

где K – раундовый ключ, который генерируется для каждого из 13 раундов генерируется по формуле:

$$K_i = LPS(K_{i-1} \oplus C_{i-1}), i \in 2, \ldots, 13 \tag{4}$$

где C – итерационная константа, значения которой указаны в ГОСТ 34.11- 2012.

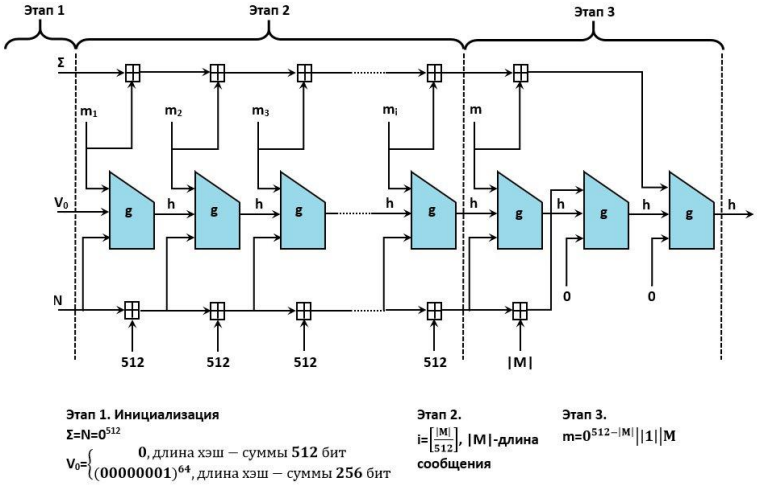

Рис. 1. Схема вычисления хэш-функции

Хэширование считается в три этапа. Первый этап — инициализация всех нужных параметров, второй этап представляет собой итерационную конструкцию Меркла — Дамгорда с процедурой МД-усиления, третий этап — завершающее преобразование: функция сжатия применяется к сумме всех блоков сообщения и дополнительно хэшируется длина сообщения и его контрольная сумма.

Различные элементы алгоритма хэширования были реализованы на языке VHDL и представлены в виде модулей:

«check_module» - предназначен для контроля расчета хэшфункции «Стрибог» и проверки на совпадение рассчитанного значения с контрольным значением;

«tb» - предназначен для ввода входных данных и моделирования тактовой частоты.

«gh_fixed512_logic» предназначен для расчета хэш-функции «Стрибог».

«gh_round_efunc_v2_logic» предназначен для расчета функции блочного шифра.

«gh_round_logic» также предназначен для расчета функции блочного шифра.

«gh_round_lps_logic» предназначен для организации связей между модулями расчета L – преобразования, P – преобразования, S – преобразования.

«gh_round_ltran_logic» предназначен для расчета L – преобразования (линейное преобразование).

«gh_round_perm_logic» предназначен для расчета P – преобразования (преобразование перестановки).

«gh_round_subs_logic» предназначен для расчета S – преобразования (преобразование подстановки).

На рисунке 2 демонстрируется значение, полученное посредством применения смоделированной хэш-функции, которое совпадает с контрольным примером, представленным в [6].

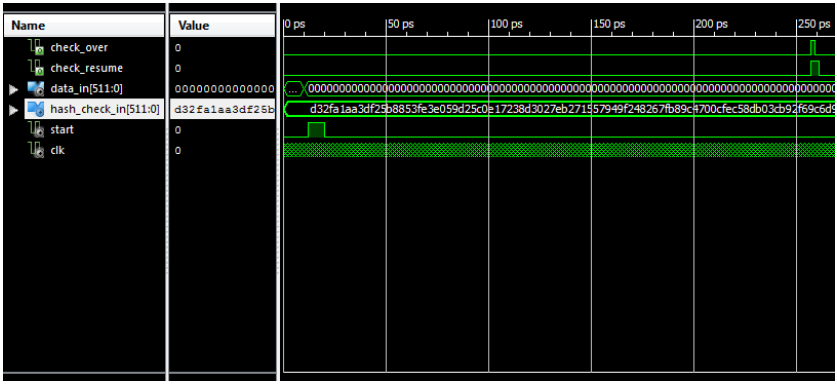

Рис. 2. График выходных сигналов

Заключение

Таким образом, реализованный модуль контроля целостности данных на ПЛИС, согласно результатам моделирования, выполняет свою функцию. Модуль рассчитывает значение хэш-функции и сверяет с исходным значением, после чего информирует о результате сравнения. Был описан алгоритм работы модуля и результат моделирования.

Список литературы

1. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. Учебное пособие. Изд. 2-е, Гелиос АРВ, 2005. – 480 с.

2. Диченко, С. А. Гибридный крипто-кодовый метод контроля и восстановления целостности данных для защищённых информационно-

аналитических систем / С. Диченко, О. Финько // Вопросы кибербезопасности. – 2019. – № 6(34). – С. 17-36.

3. Диченко, С. А. Контроль и обеспечение целостности информации в системах хранения данных / С. Диченко // Наукоемкие технологии в космических исследованиях Земли. – 2019. – Т. 11. – № 1. – С. 49-57.

4. Левин, В. Ю. О повышении криптостойкости однонаправленных хэш-функций / В. Ю. Левин // Фундамент. и прикл. матем. – 2009. – Т. 15. Выпуск 5. – С. 171-179.

5. Dichenko, S. Two-dimensional control and assurance of data integrity in information systems based on residue number system codes and cryptographic hash functions / S. Dichenko, O. Finko // Integrating Research Agendas and Devising Joint Challenges International Multidisciplinary Symposium ICT Research in Russian Federation and Europe. – 2018. – P. 139-146.

6. ГОСТ 34.11-2012 «Информационная технология. Криптографическая защита информации. Функция хэширования».

7. Диченко, С. А. Разработка алгоритма контроля и обеспечения целостности данных при их хранении в центрах обработки данных / С. А. Диченко [и др.] // Сб. науч. статей VIII Междунар. молод. научнопр. конф. с элементами науч. шк. – Омск: Омский ГТУ, 2018. – С. 40-43.

8. Диченко, С. А. Системный анализ проблемы обеспечения целостности данных в информационно-аналитических системах / С. А. Диченко // Информатика: проблемы, методы, технологии. Сборник материалов XX Международной научно-методической конференции. Под ред. А.А. Зацаринного, Д.Н. Борисова. – Воронеж, 2020. – С. 1001-1005.

9. Диченко, С. А. Алгоритм проверки достоверности контрольной информации, используемой при обеспечении целостности данных в условиях деструктивных воздействий злоумышленника и среды / С. А. Диченко // Информатика: проблемы, методы, технологии. Сборник материалов XX Международной научно-методической конференции. Под ред. А.А. Зацаринного, Д.Н. Борисова. – Воронеж, 2020. – С. 996-1000.

10. Диченко, С. А. Концептуальная модель обеспечения целостности информации в современных системах хранения данных / С. А. Диченко // Информатика: проблемы, методология, технологии. Сборник материалов XIX международной научно-методической конференции. Под ред. Д. Н. Борисова. – Воронеж, 2019. – С. 697-701.

11. Хэмминг, Р. В. Теория кодирования и теория информации: Пер. с англ. – М.: Радио и связь, 1983. – 176 с.