УДК 656.072

ИСПОЛЬЗОВАНИЕ ХЭШ-ФУНКЦИЙ ДЛЯ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ОТ НЕСАНКЦИОНИРОВАННОГО ДИСТАНЦИОННОГО УПРАВЛЕНИЯ МЕХАНИЗМОМ

Троянов Я.И., студент гр. МРб-121, 4 курс
Научный руководитель: Чичерин И.В., к.т.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева,
г. Кемерово

В ходе автоматизации технологического процесса, так или иначе, приходится сталкиваться с использованием телемеханики. Телемеханика – это совокупность методов и технических средств, предназначенных для передачи и приема информации с целью дистанционного управления оборудованием и контроля его работы. Одним из каналов передачи информации может служить радиоканал. Телемеханику, в которой для передачи команд управления и контроля используют каналы радиосвязи, называют радиотелемеханикой. При работе по радиоканалу, всегда стараются добиться: обеспечения высокой надёжности передачи команд управления; сокращения времени запаздывания сигналов; высокой степени автоматизации процессов сбора информации. Но гораздо реже ставят цель обеспечения безопасности передачи информации.

На первый взгляд это может показаться не столь важным пунктом, в перечне основных целей, но если задуматься о возможности умышленного искажения информации и подачи ложных команд, то станет ясно насколько данная проблема актуальна. Сейчас большинство предприятий автоматизировано и многие из них управляют своим технологическим процессом посредством радиотелемеханики, слабая защищенность которой, может привести к неправильной работе предприятия или даже к аварии. Таким образом, защита от случайного или преднамеренного воздействия на систему телемеханики, становится очень важной частью разработки системы.

Одним из способов защиты, может служить использование распространенной технологии хэширования сообщений.

Хэш-функции широко применяются для обеспечения безопасности, например в электронной цифровой подписи, или аутентификации. Основная задача хэш-функций – генерация дайджестов, уникальных для конкретного документа, то есть в получении из сообщения произвольного размера, значения фиксированного размера. При этом изменяя исходные данные хоть на один бит, мы получим совсем другое число на выходе.

Основные свойства хэш-функции:

1. На вход хэш-функции подается сообщение произвольной длины;

2. На выходе хэш-функции формируется блок данных фиксированной длины;

3. Значения на выходе хэш-функции распределены по равномерному закону;

4. При изменении одного бита на входе хэш-функции существенно изменяется выход.

Для обеспечения устойчивости хэш-функции к атакам она должна удовлетворять следующим требованиям:

1. Если мы знаем значение хэш-функции h, то задача нахождения сообщения M такого, что H(M) = h, должна быть вычислительно невозможной;

2. При заданном сообщении M задача нахождения другого сообщения M’, такого, что H(M) = H(M’), должна быть вычислительно невозможной.

Хэш-функции строятся на многократном повторении математических действий, когда исходное сообщение раскладывается на отдельные блоки фиксированного размера, и над этими блоками выполняются преобразования. Часто в ходе преобразования, блоки сжимаются, так как выход обычно меньше по размеру, чем блок, подаваемый на вход.

На рисунке 1 рассмотрим один из возможных алгоритмов обеспечения безопасности при телеуправлении удаленным механизмом по радиоканалу.

Рисунок 1. Алгоритм дистанционного управления механизмом

После анализа предложений Программируемых логических контроллеров (ПЛК) на рынке Российской Федерации наш выбор был остановлен на ПЛК110, производимом Российской компанией ОВЕН. Кроме привлекательной цены огромную роль в выборе сыграло наличие бесплатного локализованного программного обеспечения «CoDeSys» и весьма развитой службы поддержки производителем. На сайте производителя www.owen.ru имеется множество примеров работающих программ «на все случаи жизни». Кроме того, работает форум, на котором разработчики активно обмениваются опытом и бескорыстно помогают начинающим. Ну и наконец, написанная на CoDeSys программа может быть легко перенесена на любую другую платформу.

На основе сравнительной оценки хэш-функций, проведенной пользователем madcomaker и представленной на интернет ресурсе www.habrahabr.ru, была выбрана хэш-функция «Ly», из-за наиболее равномерного распределения значений.

В связи с ограниченной функциональностью языков технологического программирования стандарта МЭК61131-3 для программируемых логических контроллеров, целесообразно практически убедиться в возможности программной реализации хэш-функций для защиты радиотелемеханики. Для проверки возможности выполнения вышеприведенного алгоритма на программируемых логических контроллерах ОВЕН ПЛК110 была написана программа на языке технологического программирования «ST». Написание программы осуществлялось в среде программирования «CoDeSys v.2.3», являющейся штатным средством разработки программ не только для контроллеров ОВЕН, но и для контроллеров других производителей.

Ниже приведен листинг программы, на котором step1 и step3 – запросы контроллера Пункта управления (ПУ), а step2 и step4 – ответы контроллера Контролируемого пункта (КП):

```
1.  step1[i].AdrCorr:=2;                    (* Адрес кому *)
2.  step1[i].MyAdr:=1;                      (* Адрес от кого *)
3.  step1[i].Cmd:=4;                        (* Номер команды *)
4.  step1[i].Meh:=16#0001;                  (* Номер механизма *)
5.  step1[i].Num:=16#FFFF;                  (* Состояние механизма *)
6.  Req:=hash_Ly(ADR(OldReq),SystemTime);   (* Уникальный запрос *)
7.  OldReq:=Req;                            (* Запоминаем запрос *)
8.  step2[i].AdrCorr:=1;                    (* Адрес кому *)
9.  step2[i].MyAdr:=2;                      (* Адрес от кого *)
10.step2[i].Cmd:=16#84;                     (* Номер команды *)
11.step2[i].Dat:=Req;                       (* Запрос на подтверждение *)
12.Req:=hash_Ly(ADR(OldReq),4041984);       (*Готовим ответ*)
13.step3[i].AdrCorr:=2;                     (* Адрес кому *)
14.step3[i].MyAdr:=1;                       (* Адрес от кого *)
15.step3[i].Cmd:=3;                         (* Номер команды *)
16.step3[i].Dat:=Req;                       (* Подтверждение *)
```

```
17.step4[i].AdrCorr:=1;          (* Адрес кому *)
18.step4[i].MyAdr:=2;            (* Адрес от кого *)
19.step4[i].Cmd:=16#83;          (*Номер команды *)
20.step4[i].Meh:=16#0001;        (* Номер механизма *)
21.step4[i].Num:=16#FFFF;        (* Состояние механизма *)
```

Программа генерирует запросы и ответы, передаваемые контроллерами. Результатом работы программы является заполнение массивов структур, содержащих текст запросов и ответов. На рисунке 2 приведено изображение, полученное в результате эмуляции работы контроллера в подпрограмме визуализации CoDeSys. На нем видно, как два контроллера последовательно обмениваются двадцатью запросами и ответами.

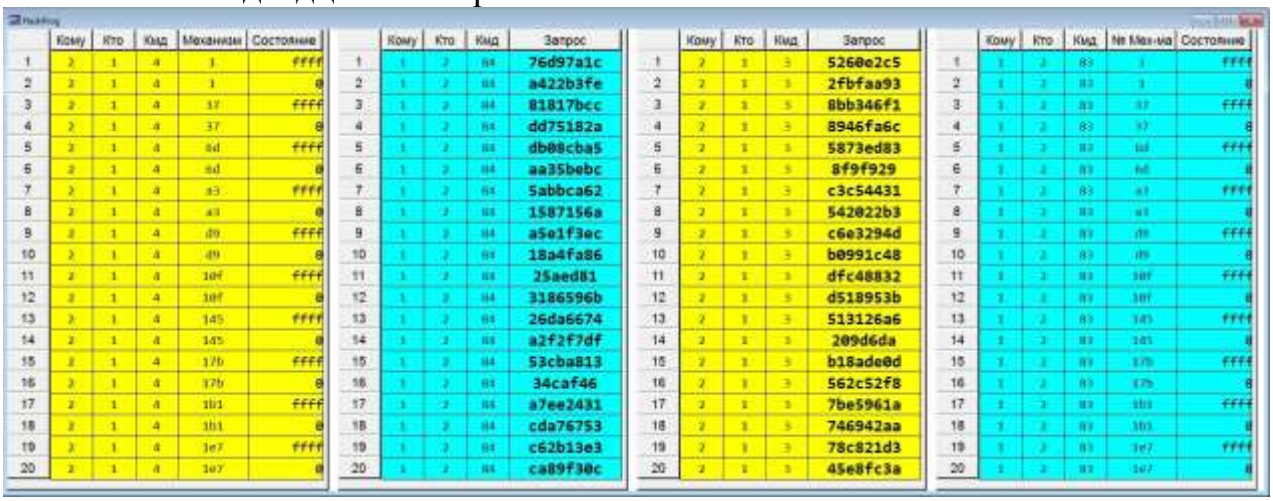

Рисунок 2. Окно программы отображения запросов и ответов 2 контроллеров

На рисунке 3 приведено распределение значений пятидесяти идущих подряд значений запросов и ответов.

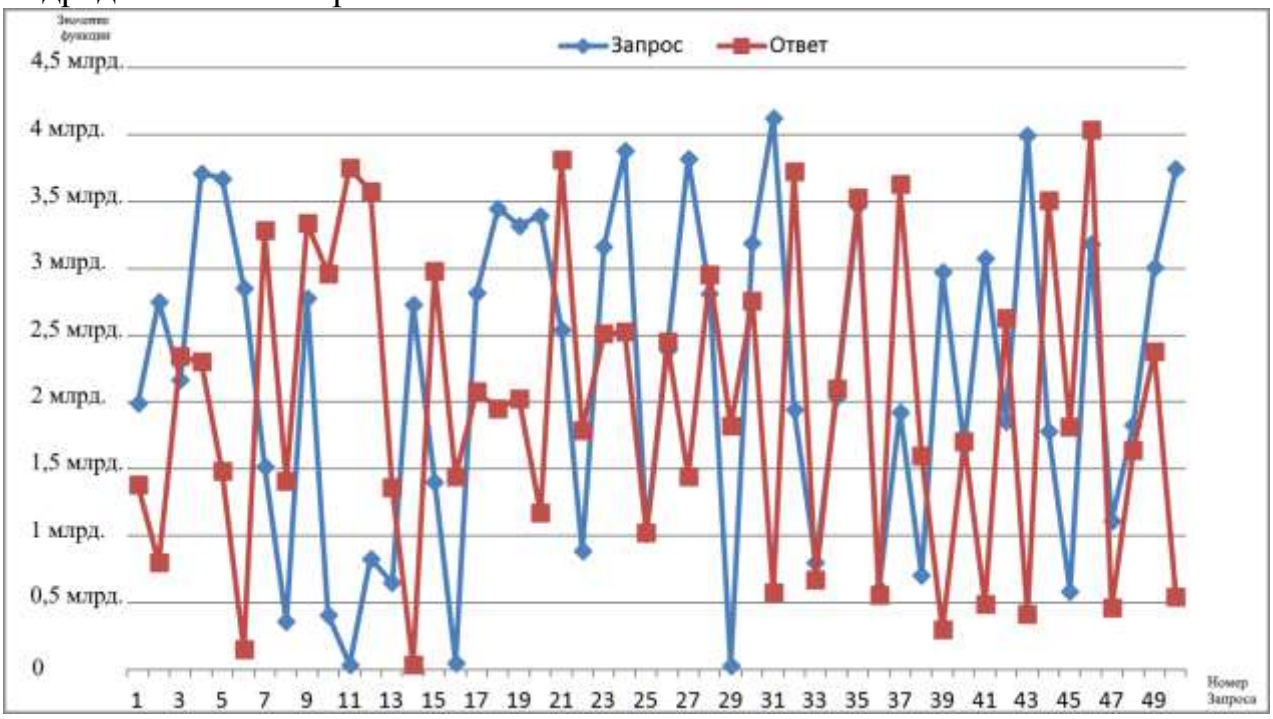

Рисунок 3. Распределение числовых значений запросов и ответов.

Анализ графика позволяет утверждать, что все сгенерированные программой значения случайны и непредсказуемы, а так же то, что запросы не коррелируют с ответами.

Листинг хэш функции «Ly»

```
1. hash_Ly:=Start;
2. FOR i:=0 TO 3 DO
3.     hash_Ly := (hash_Ly * 1664525) +*(pt^) + 1013904223;
4.     pt:=pt+1;
5. END_FOR
6. FOR i:=0 TO 7 DO
7.     hash_Ly := (hash_Ly * 1664525) +Ly[i] + 1013904223;
8. END_FOR
```

Выводы:

1. Внешне обмен между контроллерами выглядит как передача команд управления и получение подтверждения их выполнения.
2. Благодаря применению технологии хэширования удалось сформировать случайные, не повторяющиеся и равномерно распределенные в заданном диапазоне значения запросов на подтверждение команды, что показало верность идеи использования технологии хэширования.
3. Генерация ответа (подтверждения) с применением неизвестного злоумышленнику начального состояния хеш-функции удалось получить внешне не корреолирующий с запросом ответ, который невозможно «угадать», так как используется 32 битный хэш, то вероятность выбора валидного ответа на запрос составляет 1 на 4 294 967 296 (2^{32}) или $2{,}3 \times 10^{-9}$. Перебор всех комбинаций для одного запроса, при скорости обмена данными 1 раз в секунду займет 136 лет.
4. При необходимости стойкость к взлому может быть значительно увеличена путем введения дополнительных итераций. Например, введение еще одной итерации увеличивает время подбора до 60 миллиардов лет. (Справочно: время существования Земли около 4,54 миллиарда лет)

Список литературы:

1. Криптографические методы защиты информации. Хэш-функции http://www.volpi.ru/umkd/zki/index.php?man=1&page=20
2. Несколько простых хэш-функций и их свойства https://habrahabr.ru/post/219139/
3. Программирование программируемых логических контроллеров ОВЕН ПЛК110 и ПЛК160 Руководство пользователя Версия 1.9.

4. Руководство пользователя по программированию ПЛК в CoDeSys 2.3 Редакция RU 2. 8, для CoDeSys V2. 3.9.x