ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ВСТРАИВАНИЯ ОТЕЧЕСТВЕННЫХ КРИПТОГРАФИЧЕСИХ АЛГОРИТМОВ ЗАЩИТЫ ИНФОРМАЦИИ В СОВРЕМЕННЫЕ ПРОГРАММНЫЕ СРЕДЫ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Веремейчик Д. В.

Скудняков Ю. А. – канд. техн. наук, доцент

В настоящее время в недостаточной степени применяются разработанные в Республике Беларусь криптографические методы защиты информации. Например, алгоритмы, определенные в СТБ 34.101.31 и СТБ 34.101.45. В связи с этим возникает задача исследования и осуществления процесса встраивания данных алгоритмов защиты информации в современные программные среды.

Системами, позволяющими эффективно внедрять и интегрировать алгоритмы защиты информации, являются криптографические библиотеки [1]. Для исследования и анализа библиотек в [1] была выбрана библиотека OpenSSL.

В докладе рассматривается добавление алгоритма имитозащиты (MAC), который определен в СТБ 34.101.31 в модуль, описанный в [1].

Внедрения MAC отличается от внедрения алгоритма шифрования или хэширования, так как в OpenSSL нет встроенного типа для описания алгоритма MAC в обобщенном интерфейсе EVP. Поэтому для его реализации использовались две структуры:

1) EVP_MD – содержит непосредственно алгоритмы необходимые для функционирования MAC;
2) EVP_PKEY_METHOD - служит для получения ключа и последующего его передачи в EVP_MD через функцию pkey_belt_mac_ctrl.

Такая архитектура была выбрана, потому что при использовании EVP_PKEY_METHOD есть возможность принять ключ для алгоритма MAC, а EVP_MD хорошо подходит для реализации самих функций выработки MAC.

Далее часть алгоритма MAC, реализованная как EVP_MD, регистрируется в системе аналогично алгоритму шифрования.

Часть, выполненная в виде EVP_PKEY_METHOD, регистрируется иначе из-за внутренних ограничений библиотеки OpenSSL.

Пример регистрации EVP_PKEY_METHOD:

```
int register_pmeth_belt(int id, EVP_PKEY_METHOD **pmeth, int flags) {
  *pmeth = EVP_PKEY_meth_new(id, flags);
  if (!*pmeth) return 0;

  if (id == belt_imit.type) {
        EVP_PKEY_meth_set_ctrl(*pmeth, pkey_belt_mac_ctrl,
                               pkey_belt_mac_ctrl_str);
        EVP_PKEY_meth_set_signctx(*pmeth, pkey_belt_mac_signctx_init,
                                  pkey_belt_mac_signctx);
        EVP_PKEY_meth_set_keygen(*pmeth, NULL, pkey_belt_mac_keygen);
        EVP_PKEY_meth_set_init(*pmeth, pkey_belt_mac_init);
        EVP_PKEY_meth_set_cleanup(*pmeth, pkey_belt_mac_cleanup);
        EVP_PKEY_meth_set_copy(*pmeth, pkey_belt_mac_copy);
        return 1;
  } else {
        /*Unsupported method*/
        return 0;
  }
}
```

где передаваемый в функцию id – это ‘nid’, полученный и зарегистрированный в системе функцией OBJ_create.

Далее в этой функции происходит регистрация функций необходимых для функционирования EVP_PKEY_METHOD, наиболее важными из которых являются pkey_belt_mac_ctrl и pkey_belt_mac_ctrl_str, служащие для получения ключа и передачи его в часть MAC, реализовованную в виде EVP_MD.

После регистрации алгоритмы, определенные в Engine, становятся доступными для использования и отображаются в списке стандартных алгоритмов OpenSSL (рисунок 1).

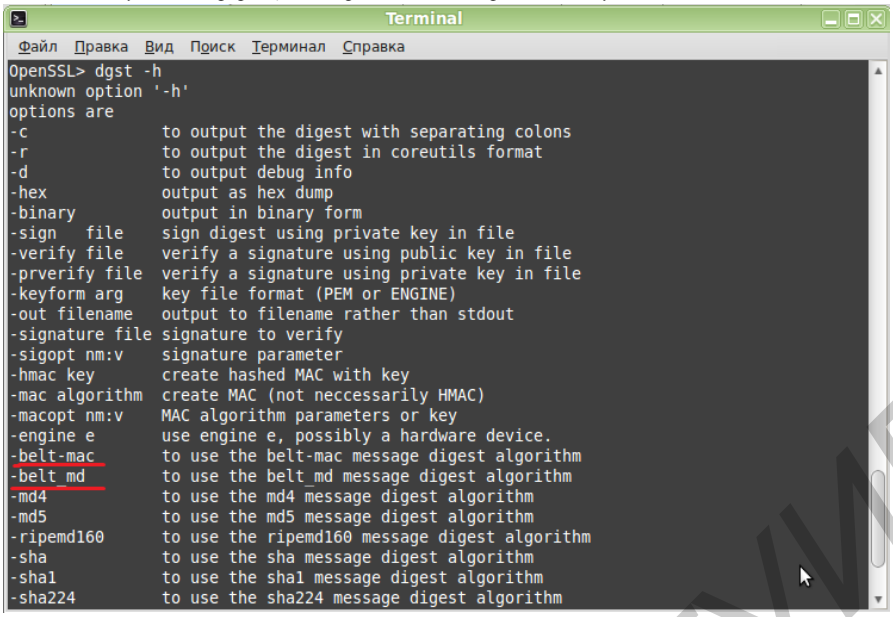

Рис.1 – Алгоритм хэширования HBELT (belt_md) и EVP_MD часть алгоритма BELT MAC (belt-mac)

Проведем тестирование алгоритмов, встроенных в полученный модуль.

```
OpenSSL> dgst -mac belt-mac -macopt hexkey:E9DEE72C8F0C0FA62DDB49F46F73964706075316ED247A3739C
BA38303A98BF6 file2.in
BELT-MAC-belt-mac(file2.in)= 7260da60138f96c9
OpenSSL>
```

Рис.2 – Использование BELT MAC

На рисунке 2 проведено тестирование алгоритма BELT MAC на тесте A.17 из стандарта СТБ 34.101.31. Файл file2.in содержит открытый текст, определенный в тесте A.17. На рисунке 3 проведено тестирование алгоритма шифрования BELT в режиме счетчика на тесте A.16 из стандарта СТБ 34.101.31..

```
OpenSSL> enc -belt-ctr -K E9DEE72C8F0C0FA62DDB49F46F73964706075316ED247A3739CBA38303A98BF6
 -iv BE32971343FC9A48A02A885F194B09A1 -in text.in -out text.out
OpenSSL>
```

Рис.3 – Использование алгоритма шифрования BELT в режиме счетчика

Параметр –K содержит ключ шифрования, параметр –iv содержит вектор инициализации, файл text.in содержит открытый текст, определенные в тесте A.16. Файл text.out содержит результат шифрования, и он совпадает с результатом, описанным в тесте A.16.

Таким образом, было произведено встраивание в OpenSSL алгоритма имитозащиты, определенного в стандарте СТБ 34.101.31, а также осуществлена проверка встроенных алгоритмов на тестах из данного стандарта. Полученные результаты совпали с результатами, описанными в СТБ 34.101.31.

Предложенный в докладе [1] и в настоящем докладе модуль адаптации, внедрения и встраивания предполагается дополнить алгоритмами, изложенными в СТБ 34.101.45. Это алгоритмы генерации и проверки параметров эллиптической кривой, генерации и проверки ключей, выработки и проверки электронной цифровой подписи.

Список использованных источников:

1 Веремейчик Д. В. Исследование и адаптация криптографических методов защиты информации в программных средах // В наст. сборнике.

2 Pravir Chandra, Matt Messier, John Viega. Network security with OpenSSL. O'Reilly, 2002. – 384 p.

3 Peter Gutmann. Cryptlib Security Toolkit Version 3.4. 2010. – 340 p.

4 Tom St Denis. LibTomCrypt Developer Manual. Ottawa, 2006. – 211 p.

5 Государственный стандарт Республики Беларусь СТБ 34.101.31-2011. Информационные технологии. Защита информации Криптографические алгоритмы шифрования и контроля целостности.