ISSN 2181-7324

ЎзМУ ХАБАРЛАРИ

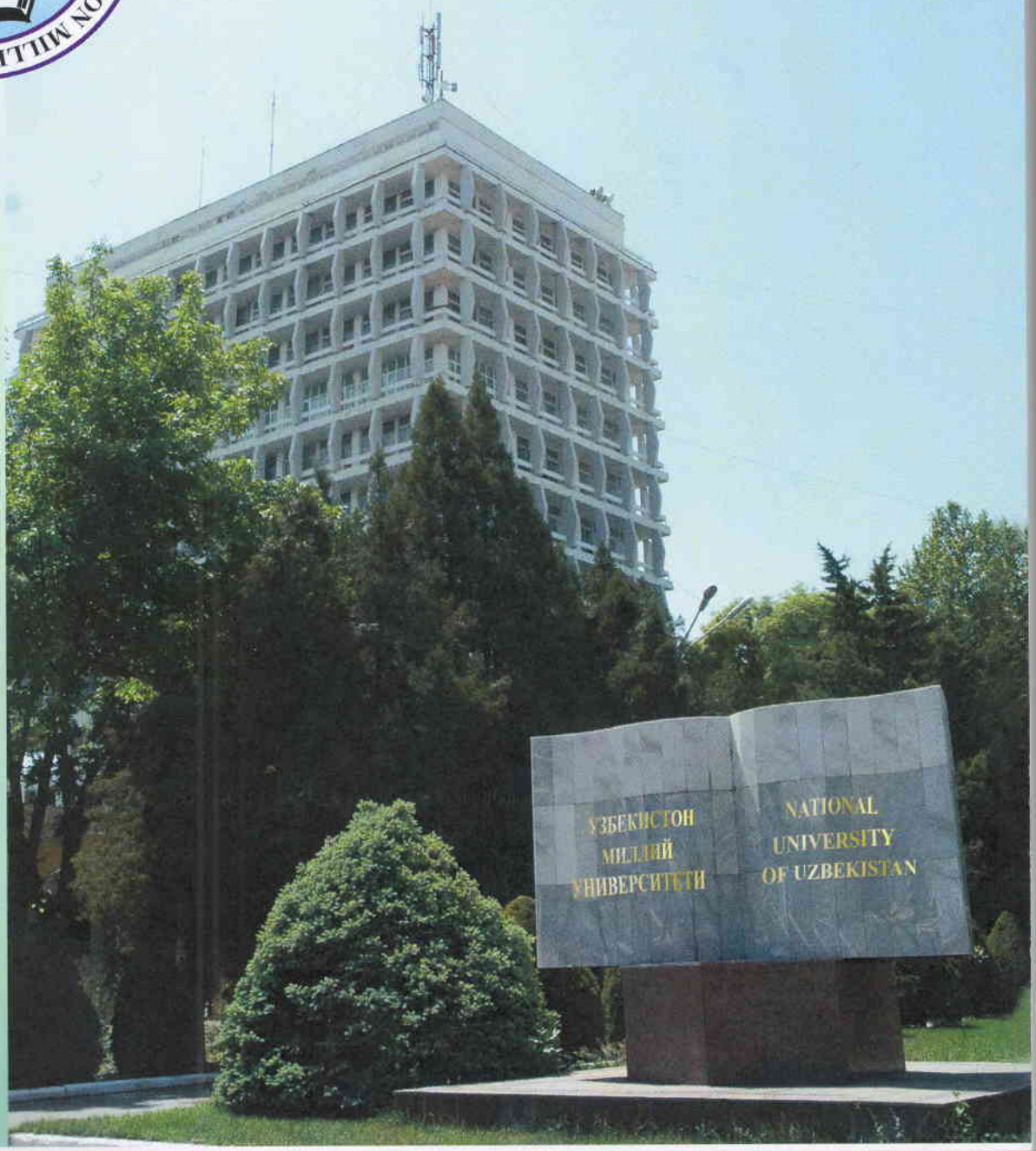
№ 2/2 ✡ 2016

MIRZO ULUG'BEK NOMIDAGI O'ZBEKISTON MILLIY UNIVERSITETI

Аниқ фанлар йўналиши

Точные науки

Exact sciences



ВЕСТНИК НУУз ✡ ACTA NUUz

ЎзМУ ХАБАРЛАРИ

ВЕСТНИК НУУз

ACTA NUUz

МИРЗО УЛУҒБЕК НОМИДАГИ ЎЗБЕКИСТОН МИЛЛИЙ УНИВЕРСИТЕТИНИНГ ИЛМИЙ ЖУРНАЛИ

ЖУРНАЛ 1997 ЙИЛДАН ЧИҚА БОШЛАГАН

2016
2/2
Аниқ фанлар

ТОШКЕНТ — 2016

МУНДАРИЖА

Математика

Информатика

Физика

СРЕДСТВО КРИПТОГРАФИЧЕСКОЙ ЗАЩИТЫ ИНФОРМАЦИИ NUUZ-CSP

Алоев Р. Д., Алаев Р. Х., Нуруллаев М. М. *

РЕЗЮМЕ

Настоящая работа содержит сведения о назначении программы, области применения, применяемых методах, ограничениях для применения средства криптографической защиты информации NUUz-CSP (далее СКЗИ NUUz-CSP) разработанного авторами.

Ключевые слова: Информационная безопасность, Криптография, CSP.

Введение. Средство криптографической защиты информации NUUz-CSP [1, 2] предназначено для создания ключей шифрования, закрытых и открытых ключей электронной цифровой подписи (ЭЦП), создания и подтверждения подлинности ЭЦП, хэширования, шифрования и имитозащиты данных с использованием алгоритмов, описанных в [3, 4, 5, 7, 8, 9].

СКЗИ NUUz-CSP может использоваться в сетях телекоммуникаций, информационных системах общего пользования, государственных корпоративных информационных системах путем встраивания в прикладные приложения, обеспечивающие хранение, обработку и передачу информации, не содержащей сведений, отнесенных к государственным секретам, а также при обмене информацией и обеспечении юридической значимости электронных документов.

СКЗИ NUUz-CSP выполняет следующие основные функции:

- формирование ключей для осуществления и проверки электронной цифровой подписи (ЭЦП) по алгоритмам 1 и 2 O‘z DSt 1092:2009 [5] и алгоритму ГОСТ Р 34.10-2001 [9];
- хэширование областей оперативной памяти и других данных по алгоритму 1 с параметром p = 256 O‘z DSt 1106:2009 [4] и алгоритму ГОСТ Р 34.11-94 [8];
- формирование и проверку результата ЭЦП в соответствии с алгоритмами 1 и 2 O‘z DSt 1092:2009 [5] и алгоритму ГОСТ Р 34.10-2001 [9];
- формирование ключей шифрования для алгоритма шифрования данных O‘z DSt 1105:2009 [3] и алгоритму ГОСТ 28147-89 [7];
- шифрование областей оперативной памяти и других данных в соответствии с алгоритмом шифрования данных O‘z DSt 1105:2009 [3] и алгоритму ГОСТ 28147-89 [7];
- работу с ключевой информацией, хранящейся на внешних носителях.

В СКЗИ NUUz-CSP предусмотрена поддержка большинства используемых в настоящее время криптографических алгоритмов.

СКЗИ NUUz-CSP может использоваться в качестве криптопровайдера по умолчанию для операционной системы Windows, помимо поддержки криптографических алгоритмов

*Алоев Р. Д.[1], Алаев Р. Х.[2] – Национальный университет Узбекистана
Нуруллаев М. М.[3] – Бухарский инженерно-технологический институт
E-mails: [1]aloevr@mail.ru, [2]mr.ruhillo@gmail.com, [3]mirxon@mail.ru

Республики Узбекистан и российских криптографических алгоритмов, СКЗИ NUUz-CSP также поддерживает некоторые общепринятые и использующиеся в ОС Windows криптографические алгоритмы, такие как RSA, 3DES, SHA-1 и т.п.

1.1. Функции работы с ключевой информацией

Изделие NUUz-CSP работает с ключевой информацией в хранилищах — ключевых контейнерах (Key Container).

Так как СКЗИ NUUz-CSP построен в соответствии с технологией Microsoft, то в контейнере содержатся следующие ключи:

• ***AT_KEYEXCHANGE*** - ключ, используемый для шифрования и обмена сессионными ключами;

• ***AT_SIGNATURE*** - ключи, используемые для создания и проверки цифровой подписи.

Примечание. Private Keys (ключи шифрования и секретные ключи подписи), содержащиеся в контейнере защищены с использованием ключа защиты, который является производным значением от значения пин-кода пользователя токена.

Вызов криптографических процедур в СКЗИ NUUz-CSP осуществляется с помощью интерфейса PKCS#11.

Основополагающими понятиями интерфейса PKCS#11 являются слот и токен. Токен является хранилищем некоторой персональной информации (различных ключей, сертификатов, приватных данных и т.п.), а слот выступает в роли связующего звена между компьютером и токеном, допускающим подключение различных токенов в различное время.

Для ***AT_KEYEXCHANGE*** и ***AT_SIGNATURE*** могут использоваться как одни и те же, так и различные слоты PKCS#11.

СКЗИ NUUz-CSP поддерживает работу с контейнерами находящимися как на жестком диске компьютера, так и на сменных носителях типа Flash Memory и Smart Card.

Примечание. Данная версия СКЗИ NUUz-CSP поддерживает работу только с Smart Card (USB Token) типа — eToken PRO 72K (Java).

Каждый контейнер имеет уникальное имя, состоящее из префикса или нескольких префиксов и самого имени. Префиксы в имени контейнера отделяются друг от друга символом "\". В имени контейнера может быть от нуля до трех префиксов:

$$Container\ Name = [pref1\backslash][pref2\backslash][pref3\backslash]Name$$

Определение расположения носителя осуществляется по первому префиксу в имени контейнера, в зависимости от наличия в функции ***CPAcquireContext*** флага ***CRYPT_MACHINE_KEYSET.*** Префиксы и место расположения носителя представлены в таблице 1.

Таблица 1 — Префиксы и место расположения носителя

Префикс	**Флаг CRYPT_MACHINE_KEYSET**	**Расположение**
отсутствует	нет	$[CSIDL\_APPDATA] \setminus uz\_csp\_containers$
отсутствует	да	$[CSIDL\_COMMON\_APPDATA] \setminus uz\_csp\_containers$
HD	нет	$[CSIDL\_APPDATA] \setminus uz\_csp\_containers$
HD	да	$[CSIDL\_COMMON\_APPDATA] \setminus uz\_csp\_containers$
FM	нет	$[FIRST\_FM] \setminus uz\_csp\_containers$
FM	да	не поддерживается
SC	нет	$[FIRST\_SC]$
SC	да	не поддерживается
[DRV]	нет	$[DRV] \setminus uz\_csp\_containers$
[DRV]	да	не поддерживается

где:

— **[CSIDL_APPDATA]** и **[CSIDL_COMMON_APPDATA]** — системные переменные;

— **[FIRST_FM]** — первый найденный носитель типа Flash Memory;

— **[FIRST_SC]** — первая найденная смарт - карта;

— **[DRV]** — явно заданный логический диск (например, ***D, E*** и т.п.).

В имени контейнера вторым префиксом является ссылка на слот для ***AT_KEYEXCHANGE***, а третьим - для ***AT_SIGNATURE***. Если третий префикс отсутствует, то ***AT_KEYEXCHANGE*** и ***AT_SIGNATURE*** хранятся на одном и том же слоте.

Защита private-объектов токена осуществляется при помощи криптографического интерфейса PKCS#5. Данный алгоритм решает сразу две задачи: зашифровывание private-данных и их защиту от случайных или преднамеренных искажений.

1.2. Функция шифрования

Функция шифрования представляет собой криптографический алгоритм, представляющий собой биективное отображение из конечного множества открытых текстов в конечное множество шифрованных текстов, в котором функция отображения зависит от секретного параметра, называемого ключом.

Функция шифрования используется для зашифрования и расшифрования информации.

В NUUz-CSP реализованы алгоритмы шифрования в соответствии с O'z DSt 1105:2009 [3], ГОСТ 28147-89 [7].

Функция шифрования [3] может использовать криптографические ключи длиной 256 или 512 бит для зашифрования и расшифрования блоков данных длиной 256 бит.

Функция шифрования используется для криптографической защиты данных, хранимых и передаваемых в сетях ЭВМ, телекоммуникаций, в отдельных вычислительных

комплексах или в ЭВМ предприятий, организаций и учреждений.

В симметричных криптосистемах обмен данными происходит в три этапа:

1) отправитель сообщения передает получателю ключ шифрования (или/и функциональный ключ) по защищенному каналу, который известен только им;

2) отправитель, с помощью ключа шифрования и функционального ключа преобразует исходные данные в зашифрованные данные и отправляет их получателю по каналу связи;

3) получатель, получив зашифрованные данные, расшифровывает их с помощью ключа шифрования и функционального ключа. Обе стороны могут воспользоваться этими ключами несколько раз.

Кроме защиты данных, функция шифрования может использоваться для защиты самих симметричных ключей при их передаче по незащищенным каналам связи. В этом случае передаваемый симметричный ключ шифруется с помощью некоторого другого ключа, называемого ключом защиты.

Функция шифрования [3] содержит в себе два режима:

— режим электронной кодовой книги (ECB);

— режим сцепления блоков (CBC).

Режим электронной кодовой книги представляет собой режим шифрования, в котором все блоки открытого текста шифруются независимо друг от друга на одном ключе, в соответствии с алгоритмом шифрования данных.

Режим ECB обычно используется при шифровании симметричных ключей.

Режим сцепления блоков представляет собой режим шифрования, в котором каждый зашифрованный (расшифрованный) блок зависит от предыдущего зашифрованного (расшифрованного) блока. Для первого блока в качестве предыдущего блока используется вектор инициализации. В случае если последний блок текста является не полным, он дополняется до необходимой длины. Эта процедура называется *паддингом (padding)*. Режим CBC обычно используется при шифровании данных.

Назначение данных функции, функциональные возможности, алгоритмы функционирования представлены в документе "O'z DSt 1105:2009. Государственный стандарт Узбекистана. Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Алгоритм шифрования данных".

1.3. Функция хеширования

Функция хэширования предназначена для осуществления однонаправленного сжимающего сотображения f из множества A в множество B, на вход которого подается сообщение произвольной длины , а выходом является строка фиксированной длины $h(M)$. Применение хэширующего преобразования позволяет уменьшить избыточность входного текста.

Функция хэширования применяется в криптографических методах обработки и защиты информации, в том числе для реализации процедур электронной цифровой подписи (далее ЭЦП) при передаче, обработке и хранении информации в автоматизированных системах.

Ниже приведены базовые требования, предъявляемые к функции хэширования.

Базовые требования к криптографической хэш-функции:

— на вход функции может подаваться сообщение любой длины;

— на выходе функции получается сообщение фиксированной длины;

— хэш-функция достаточно просто вычисляется для любого сообщения;

— хэш-функция — однонаправленная функция;

— зная сообщение M, практически невозможно найти другое сообщение M', для которого $h(M) = h(M')$.

В NUUz-CSP реализованы алгоритмы хэширования в соответствии с O'z DSt 1106:2009 [4], ГОСТ Р 34.11-94 [8].

1.4. Функция подписи

Функция электронной цифровой подписи служит для формирования и подтверждения подлинности электронной цифровой подписи (ЭЦП) под заданным сообщением (электронным документом), передаваемым по незащищенным телекоммуникационным каналам общего пользования.

При получении сообщения получатель может осуществить проверку целостности сообщения переданного отправителем и проверить достоверность авторства его отправителя.

ЭЦП является электронным аналогом письменной подписи и поэтому ЭЦП может использоваться получателем или третьей стороной для удостоверения, что сообщение было действительно подписано отправителем.

В NUUz-CSP реализованы алгоритмы формирования и проверки электронной цифровой подписи в соответствии с O'z DSt 1092:2009 алгоритм 1, алгоритм 2 [5], ГОСТ Р 34.10-2001 [9].

Для формирования и подтверждения подлинности ЭЦП в соответствии с O'z DSt 1092:2009 используются два алгоритма (Алгоритм 1, Алгоритм 2):

— без сеансового ключа;

— с сеансовым ключом.

Алгоритм 2 используется в классическом (без сеансового ключа) режиме. В Алгоритме 1 предусмотрен резервный путь обнаружения подделки ЭЦП путем введения в процесс формирования ЭЦП процедуры сеансового ключа, используемого в процессе подтверждения подлинности ЭЦП.

Состав и назначение данной функции, функциональные возможности, алгоритм функционирования представлены в документе "O'z DSt 1092:2009. Государственный стандарт Узбекистана. Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Процессы формирования и проверки электронной цифровой подписи".

1.5. Функциональные ограничения на применение

Криптографический интерфейс СКЗИ NUUz-CSP реализован в соответствии со стандартом CSP, который применим только в операционной системе Windows и не применим в других операционных системах типа Linux, Mac, Unix и т.п.

Поскольку в СКЗИ NUUz-CSP реализованы криптографические алгоритмы, которые не распознаются стандартными средствами Windows, для встраивания СКЗИ

в типовые приложения Windows (MS Outlook, Internet Explorer, VPN и т.п.) требуется внести изменения в стандартное программное обеспечение ОС (advapi32.dll, cryptsp.dll, crypt32.dll, inetcomm.dll, schannel.dll, secur32.dll, mailcomm.dll и т.п.).

Внесение подобных изменений в ОС Windows может быть осуществлено различными способами. СКЗИ NUUz-CSP не осуществляет изменения в стандартном программном обеспечении ОС.

2. Описание задачи

СКЗИ NUUz-CSP реализовано в виде следующих динамических библиотек:

— ***UZCSP.DLL*** — загрузка интерфейса CSP с помощью Crypto API;

— ***UZCSPFUNC.DLL*** — загрузка интерфейса CSP напрямую;

— ***UZPKCS11.DLL*** — загрузка интерфейса PKCS#11 (интерфейс PKCS#11 для СКЗИ NUUz-CSP , далее PKCS#11);

— ***UZSCTOKEN.DLL*** — функций работы со смарт-картой через интерфейс PKCS#11;

— ***UZVTOKEN.DLL*** — функций работы с виртуальными слотами и токенами через интерфейс PKCS#11;

— ***UZCRYPTOSP.DLL*** — библиотека криптографических процедур.

Вышеуказанные библиотеки помещаются в папку WINDOWS\SYSTEM32. Также, в комплект поставки СКЗИ NUUz-CSP входят следующие модули:

— ***UZGUI.DLL*** — интерфейс для ввода пароля к ключу, генерация случайных чисел с помощью механизма "электронная рулетка";

— ***UZCSP_INTEGRAL_TEST.EXE*** — интегральный тест для CSP СКЗИ NUUz-CSP ;

— ***UZPKCS11INI.EXE*** — инициализация виртуальных слотов и токенов для интерфейса PKCS#11;

— ***UZCRYPTOTEST.EXE*** — тесты криптографических алгоритмов;

— ***UZKEYMANAGER.EXE*** — тестовая программа для получения и просмотра сертификатов;

— ***UZKM.DLL*** — динамическая библиотека для поддержки работы тестовой программы ***UZKEYMANAGER.EXE***;

— ***CSPCON.DLL*** — динамическая библиотека для поддержки операций экспорта-импорта закрытых ключей в формате PFX.

Примечание. Программа ***UZKEYMANAGER.EXE*** и две поддерживающие ее динамические библиотеки предназначены только для тестирования работы СКЗИ NUUz-CSP с сертификатами и закрытыми ключами.

2.1. Интерфейс CSP

В СКЗИ NUUz-CSP реализован криптографический интерфейс CSP в соответствии с требованиями стандарта Cryptography Service Provider (CSP) компании Microsoft.

Интерфейс CSP СКЗИ NUUz-CSP разработан для реализации в прикладных программах криптографических алгоритмов [3],[4],[5], российских алгоритмов шифрования и электронной подписи [7, 8, 9] с помощью интерфейса UZPKCS#11[6].

Интерфейс CSP СКЗИ NUUz-CSP состоит из двух динамических библиотек ***UZC-***

SP.DLL, UZCSPFUNC.DLL, а также вспомогательной тестовой программы ***UZCSP_INTEGRAL_TEST.EXE.***

Использование СКЗИ NUUz-CSP может осуществляться как напрямую, путем загрузки библиотеки ***UZCSPFUNC.DLL*** с помощью команды LoadLibrary и получения адресов криптографических функций с помощью команды GetProcAddress, либо через интерфейс CRYPTOAPI.

2.2. Интерфейс UZPKCS#11

В СКЗИ NUUz-CSP реализован криптографический интерфейс UZPKCS#11 в соответствии с требованиями стандарта PKCS#11 v2.20: Cryptographic Token Interface Standard, разработанного в RSA laboratories [6].

Интерфейс UZPKCS#11 разработан для реализации в прикладных программах криптографических алгоритмов [3],[4],[5], российских алгоритмов шифрования и электронной подписи [7, 9], и зарубежных криптографических алгоритмов шифрования и формирования хэш-функций, а так же формирования и проверки электронной подписи для возможности использования в качестве криптопровайдера по умолчанию.

Интерфейс PKCS#11 реализован в виде трех динамических библиотек ***UZPKCS11.DLL, UZSCTOKEN.DLL*** и ***UZVTOKEN.DLL***, которые помещаются в системную папку WINDOWS\SYSTEM32. Кроме того, в комплект поставки входит исполняемый файл UZPKCS11INI.EXE, предназначенный для создания виртуальных слотов и токенов для интерфейса UZPKCS#11, ***UZGUI.DLL*** — интерфейс для ввода пароля к ключу и генерации случайных чисел с помощью "электронной рулетки".

2.2.1. Механизм "электронная рулетка"

Для инициализации датчика используется источник внешней энтропии — механизм "Электронная рулетка" (биологический ДСЧ). Для реализации "электронной рулетки" используется программный модуль ***UZGUI.DLL***. При первом вызове ПДСЧ с "электронной рулетки" снимается 32 байта — вектор инициализации ПДСЧ. В настоящей реализации СКЗИ NUUz-CSP из библиотеки ***UZGUI.DLL*** используется функция — ***BioRandom***.

2.3. Библиотека криптографических процедур

Библиотека криптографических процедур реализована в виде динамической библиотеки ***UZCRYPTOSP.DLL***, предназначенной для использования в операционной системе Windows версии XP и выше. Библиотека позволяет вызывать функции выработки ключей, хэширования и электронной подписи, реализованные в соответствии с требованиями [3, 4, 5, 7, 8, 9].

Данная библиотека может использоваться как напрямую, с помощью процедуры ***LoadLibrary***, так и с помощью интерфейса UZPKCS#11.

Для проверки работоспособности указанной выше библиотеки ***UZCRYPTOSP.DLL*** подготовлена тестовая программа ***UZCRYPTOTEST.EXE***, позволяющая вызывать криптографические процедуры из библиотеки ***UZCRYPTOSP.DLL*** и получать результаты их выполнения в виде различных файлов.

2.3.1. Тестовая программа UZCRYPTOTEST.EXE

Тестовая программа ***UZCRYPTOTEST.EXE*** позволяет вызывать из библиотеки ***UZCRYPTOSP.DLL*** процедуры выработки ключей, хэширования и подписи и получать

результаты их выполнения в виде файлов.

Для использования программы ***UZCRYPTOTEST.EXE*** в папке, где находится эта программа, должна быть создана поддиректория ***UZ_TESTS***, в которой должны быть следующие поддиректории:

— ***HASH256*** — для проверки функции хэширования;

— ***SIGN_ALG1*** — для проверки алгоритма 1 ЭЦП;

— ***SIGN_ALG2*** — для проверки алгоритма 2 ЭЦП.

В поддиректории HASH256 находятся входные параметры и результаты выполнения операции хэширования при значении модуля p=256.

В ней две поддиректории:

— ***DATA*** — содержит хэшируемый текст в файле ***DATA_TO_HASH***, а в файл ***HASH*** помещается результат выполненного хэширования;

— ***PARAMS*** — содержит параметры ***KEY*** и ***ZICHLASH*** алгоритма хэширования в файлах с соответствующими именами.

Данные в файлы записываются как текстовое представление HEX-символов, за исключением файлов, содержащих данные для хэширования или подписи, в которые информация записывается в обычном текстовом представлении.

В поддиректории ***SIGN_ALG1*** находятся входные параметры и результаты выполнения операции ЭЦП по алгоритму 1.

В ней три поддиректории:

— ***DATA***— содержит подписываемый текст в файле ***DATA_TO_SIGN***, а в файл ***SIGNATURE*** помещается результат выполнения операции ЭЦП;

— ***PARAMS***— содержит параметры ***P, Q, R, G, R1*** алгоритма ЭЦП в файлах с соответствующими именами;

— ***KEYS***— содержит ключи алгоритма ***U, X, Y, Z*** алгоритма ЭЦП в файлах с соответствующими именами.

В поддиректории SIGN_ALG2 находятся входные параметры и результаты выполнения операции ЭЦП по алгоритму 2. В ней три поддиректории:

— ***DATA***— содержит подписываемый текст в файле ***DATA_TO_SIGN***, а в файл ***SIGNATURE*** помещается результат выполнения операции ЭЦП;

— ***PARAMS***— содержит параметры ***A, B, P, T, W, Xn, Yn*** алгоритма ЭЦП в файлах с соответствующими именами;

— ***KEYS***— содержит ключи алгоритма ***D, X, Y*** алгоритма ЭЦП в файлах с соответствующими именами.

Для проверки работоспособности самой библиотеки ***UZCRYPTOSP.DLL*** необходимо запустить ***UZCRYPTOTEST.EXE*** с параметрами, представленными в таблице 2.

Таблица 2 — Параметры ***UZCRYPTOTEST.EXE***

Параметр	Выполняемая операция	Примечания
GenKey GenKey - alg1	Выработка ключей для алгоритма 1 ЭЦП	Должны быть заданы параметры ***P, Q, R, G***. Результат помещается в файлы ***U, X, Y, Z***
GenKey - alg2	Выработка ключей для алгоритма 2 ЭЦП	Должны быть заданы параметры ***A, B, P, T, W, Xn, Yn***. Результат помещается в файлы ***D, X, Y***.
Hash	Хэширование текста	Должны быть заданы параметры ***KEY*** и ***ZICHLASH***, а также текст для хэширования ***DATA_TO_HASH***. Результат помещается в файл ***HASH***.
Sign Sign - alg1	Подпись текста с использованием алгоритма 1 ЭЦП без сеансового ключа	Должны быть заданы параметры и ключи для подписи по алгоритму 1, а также текст в файле ***DATA_TO_SIGN***. Результат помещается в файл ***SIGNATURE***.
Sign - alg1skey	Подпись текста с использованием алгоритма 1 ЭЦП с сеансовым ключом	Должен быть задан сеансовый ключ в файле R1.
Sign - alg1cnt	Проверка контрольного примера алгоритма 1 ЭЦП	Должна быть поддиректория ***CONTROL_EXAMPLE*** с параметрами контрольного примера
Sign - alg2	Подпись текста с использованием алгоритма 2 ЭЦП	Аналогичные к примечаниям для алгоритма 1
Verify Verify - alg1	Проверка подписи для алгоритма 1 без сеансового ключа	
Verify - alg1skey	Проверка подписи для алгоритма 1 с сеансовым ключом	
Verify - alg1cnt	Проверка подписи из контрольного примера	
Verify - alg2	Проверка подписи для алгоритма 2	

2.3.2. Импортируемые функции

Функции, импортируемые из библиотеки ***UZCRYPTOSP.DLL***, условно могут быть разбиты на следующие классы:

— функции инициализации;

— функции выработки ключей;

— функции хэширования;

— функции подписи;

— функции проверки подписи.

3. Входные и выходные данные

Криптографический интерфейс СКЗИ NUUz-CSP создан в соответствии с требованиями стандарта Cryptography Service Provider (CSP) компании Microsoft. Описание стандарта CSP можно найти на сайте Microsoft. Описание входных и выходных данных для стандарта CSP также можно найти на сайте Microsoft по нижеуказанным ссылкам.

Функции соединения с CSP:

Функция	**Описание**
CPAcquireContext	Связывает контейнер с ключами с указателем CSP.
CPGetProvParam	Выдает параметры CSP.
CPReleaseContext	Освобождает указатель, полученный CPAcquireContext.
CPSetProvParam	Устанавливает специфические параметры CSP.

Выработка ключей и функции обмена ключами CSP:

Функция	**Описание**
CPDeriveKey	Создает ключ из пароля.
CPDestroyKey	Удаляет ключ из памяти.
CPDuplicateKey	Создает копию ключа.
CPExportKey	Экспорт ключа.
CPGenKey	Выработка случайного ключа.
CPGenRandom	Выработка случайных чисел.
CPGetKeyParam	Получение параметров ключа.
CPGetUserKey	Получение указателя ключа пользователя.
CPImportKey	Импорт ключа.
CPSetKeyParam	Установка параметров ключа.

Функции зашифрования и расшифрования:

Функция	**Описание**
CPDecrypt	Расшифровывает шифрованный текст с помощью ключа для шифрования.
CPEncrypt	Зашифровывает открытый текст с помощью ключа для шифрования.

Функции хэширования и ЭЦП:

Функция	Описание
CPCreateHash	Создание указателя хэш-функции.
CPDestroyHash	Удаление указателя хэш-функции.
CPDuplicateHash	Создание копии хэш-функции.
CPGetHashParam	Получение свойств хэш-функции.
CPHashData	Хэширование данных.
CPHashSessionKey	Хэширование ключа сессии.
CPSetHashParam	Установка параметров хэширования.
CPSignHash	Подпись хэш-функции.
CPVerifySignature	Проверка подписи хэш-функции.

ЛИТЕРАТУРА

1. Алоев Р. Д., Нуруллаев М. М., Алаев Р. Х. Разработка криптографического провайдера на основе национальных стандартов. Ташкент. УзМУ хабарлари, 2013, 2, 32-35.
2. Aloev R. D. Program Structure CSP. Materials Science and Engineering Conference "Applied Mathematics and Information Security", 28-30 April, Tashkent, 2014, 302-306.
3. O'z DSt 1105:2009 - Государственный стандарт Узбекистана. Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Алгоритм шифрования данных.
4. O'z DSt 1106:2009. - Государственный стандарт Узбекистана. Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Функция хеширования.
5. O'z DSt 1092:2009. - Государственный стандарт Узбекистана. Информационная технология КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ. Процессы формирования и проверки электронной цифровой подписи.
6. Расширение PKCS#11 для использования российских криптографических алгоритмов. Москва, 2008.
7. ГОСТ 28147-89 - Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования.
8. ГОСТ Р 34.11-94 Информационная технология. Криптографическая защита информации. Функция хэширования.
9. ГОСТ Р 34.10-2001 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.

REZYUME

Mazkur maqolada mualliflar tomonidan ishlab chiqilgan axborotni kriptografik himoyalash vositasi (NUUz-CSP) ning qo'llanish sohalari, unda qo'llanilgan metodlar, asosiy xarakteristikalari va imkoniyatlari bayon etiladi.

Kalit so'zlar: Axborot xavfsizligi, Kriptografiya, CSP.

RESUME

This work contains information about the program, applied methods, restrictions for application of means of cryptographic protection of information NUUz-CSP developed by authors.

Key words: Information Security, Cryptography, CSP.