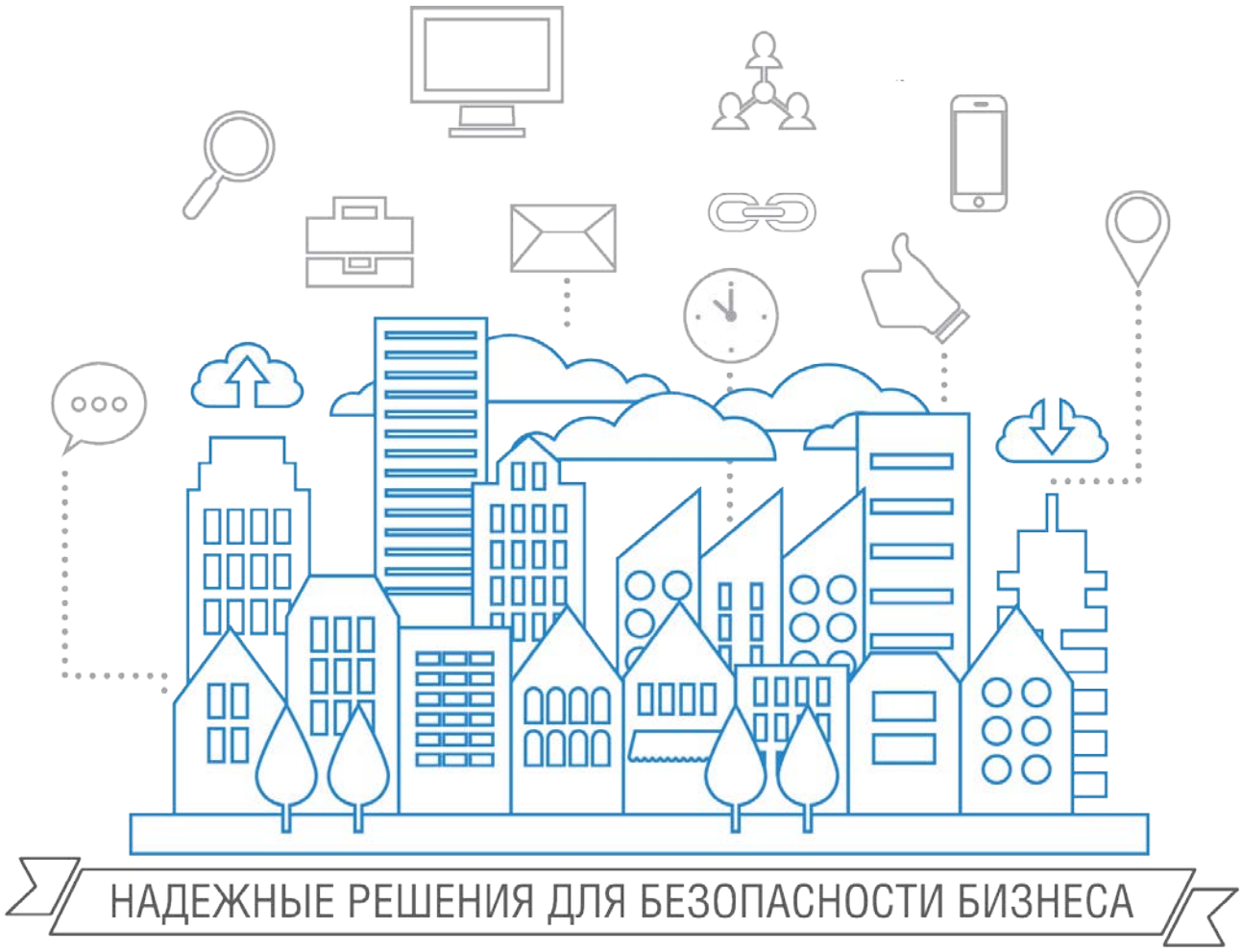
ГАЗИНФОРМСЕРВИС

198096, г. Санкт-Петербург, ул. Кронштадтская, д.10, лит. А, тел.: (812) 677-20-50, факс: (812) 677-20-51
Почтовый адрес: 198096, г. Санкт-Петербург, а/я 59, e-mail: resp@gaz-is.ru, www.gaz-is.ru
р/с 40702810800000001703 Ф-л Банка ГПБ (АО) в г. Санкт-Петербурге БИК 044030827,
к/с 30101810200000000827, ОКПО 72410666, ОГРН 1047833006099, ИНН/КПП 7838017968/783450001

Криптографическая платформа «Litoria Crypto Platform» для iOS

Руководство системного программиста



Санкт-Петербург, 2017

Аннотация

В документе приводится руководство системного программиста криптографической платформы «Litoria Crypto Platform» (далее криптоплатформа или платформа «Litoria Crypto Platform») для iOS.

В разделе «Общие сведения о криптоплатформе» описано назначение криптоплатформы, перечислены функции криптоплатформы и приведены краткие описания основных функций, указано программное обеспечение, которое необходимо для функционирования платформы «Litoria Crypto Platform», и приведены входные параметры для функций криптоплатформы и выходные параметры, которые формирует криптоплатформа.

В разделе «Структура криптоплатформы» приведены заголовки объявления, входные и выходные параметры, а также возвращаемые значения импортируемых функций платформы «Litoria Crypto Platform».

В разделе «Сообщения программисту» описан способ получения информации об ошибке и приведены коды ошибок, возвращаемые криптоплатформой.

В конце документа приведен список использованных сокращений.

Содержание

1 Общие сведения о криптоплатформе

Криптографическая платформа «Litoria Crypto Platform» для iOS предназначена для выполнения операций с сертификатами, а также операции создания и проверки электронной подписи файловых объектов на мобильных устройствах, работающих под iOS.

Криптоплатформа обеспечивает возможность обращения к функциям криптопровайдера «КриптоПро CSP» версии 3.6 R3 FrameWork для iOS.

Криптографическая платформа «Litoria Crypto Platform» для iOS выполняет следующие основные функции:

- установка личного сертификата пользователя с карточки на устройство;
- установка сертификата в заданное хранилище;
- хеширование данных по ГОСТ;
- создание отделенной электронной подписи для файла;
- проверка отделенной электронной подписи.

1.1 Описание функций криптоплатформы «Litoria Crypto Platform»

1.1.1 Создание ЭП

Электронная подпись (ЭП) – реквизит электронного документа, предназначенный для защиты данного электронного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа сертификата ключа подписи и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе.

Отделенная ЭП – файл подписи расположен отдельно от подписываемого файла.

Сертификат ключа подписи (далее – сертификат) – документ на бумажном носителе или электронный документ с электронной подписью уполномоченного лица удостоверяющего центра, которые включают в себя открытый ключ электронной подписи и которые выдаются удостоверяющим центром участнику информационной системы для подтверждения подлинности электронной подписи и идентификации владельца сертификата ключа подписи.

Для создания ЭП должен быть осуществлен выбор сертификата ключа подписи.

1.1.2 Проверка ЭП

Проверка ЭП подразумевает подтверждение подлинности ЭП в электронном документе, то есть подтверждение:

- принадлежности ЭП в электронном документе владельцу сертификата ключа;
- отсутствия искажений в подписанном данной ЭП электронном документе;
- момента ЭП;
- действительности сертификата на момент создания ЭП.

Проверка отделенной ЭП – проверка корректности самого файла подписи.

Если во время проверки сертификат подписчика не был найден в локальном хранилище, то этот сертификат берется из проверяемого файла.

1.2 Условия применения

Криптографическая платформа «Litoria Crypto Platform» для iOS функционирует под управлением операционной системы Apple iOS версии 4.2 и выше.

Для работы криптоплатформы требуется установка криптопровайдера «КриптоПро CSP» версии 3.6 R3 FrameWork для iOS.

1.3 Общая характеристика входной и выходной информации

1.3.1 Входные данные

Криптографическая платформа «Litoria Crypto Platform» выполняет функции создания и проверки ЭП для файлов произвольного типа.

1.3.2 Выходные данные

На выходе криптоплатформа «Litoria Crypto Platform» для iOS может формировать следующую информацию:

- массив байт результата операции хеширования и его размер;
- информация о проверенной ЭП;
- сообщение об ошибке (если она имеет место).

2 Структура криптоплатформы

Криптографическая платформа «Litoria Crypto Platform» для iOS реализована в виде статической платформы на языках программирования C++ и Objective-C. Классификация импортируемых функций, содержащихся в исходных кодах криптоплатформы, приведена в таблице 2.1. Подробное описание функций смотрите в соответствующих разделах.

Таблица 2.1. Классификация импортируемых функций

Название группы	Функция	Назначение	Расположение
Работа с ключевыми контейнерами	InstallCertAndKeysFromTokenToStore	Установка личного сертификата пользователя с карточки на устройство	Litoria Crypto Platform.h
	InstallCertAndKeysFromTokenToStore AndStartCachingContainer	Установка личного сертификата пользователя с карточки на устройство и кеширование контейнера закрытого ключа	
	InstallCertToStore	Установка сертификата в заданное хранилище	
Хеширование данных	HashData	Хеширование данных по ГОСТ	
Работа с ЭП	SignFile	Создание ЭП	
	VerifyFile	Проверка ЭП	

Импортируемые функции криптоплатформы могут быть вызваны из любого интерфейсного модуля.

2.1 Работа с ключевыми контейнерами

2.1.1 Функция InstallCertAndKeysFromTokenToStore()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default"))) InstallCertAndKeysFromTokenToStore(const LPSTR pinCode);

Входные параметры:

pinCode – пин-код пользователя

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).
Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

2.1.2 Функция InstallCertAndKeysFromTokenToStoreAndStartCachingContainer()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default")))
InstallCertAndKeysFromTokenToStoreAndStartCachingContainer(const LPSTR pinCode);

Входные параметры:

pinCode – пин-код пользователя

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).
Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

2.1.3 Функция InstallCertToStore()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default"))) InstallCertToStore(const BYTE* pbCertBlob, const DWORD cbCertBlob, const LPWSTR storeName);

Входные параметры:

pbCertBlob – устанавливаемый сертификат в виде набора байт
cbCertBlob – размер сертификата
storeName – имя хранилища сертификатов

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).
Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

2.2 Хеширование данных

2.2.1 Функция HashData()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default"))) HashData(const BYTE* pbContent, const DWORD cbContent, BYTE*& pbHashedData, DWORD& cbHashedData);

Входные параметры:

pbContent – массив байт для хеширования
cbContent – размер массива байт для хеширования

Выходные параметры:

pbHashedData – массив байт, результат операции хеширования
cbHashedData – размер массива байт результата операции хеширования

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).
Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

2.3 Работа с ЭП

2.3.1 Функция SignFile()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default"))) SignFile(const BYTE* pbCertBlob, const DWORD cbCertBlob, const LPSTR inFileName, const LPSTR outFileName);

Входные параметры:

pbCertBlob – массив байт сертификата пользователя
cbCertBlob – размер массива байт сертификата пользователя
inFileName – имя файла, который будет подписываться
outFileName – имя выходного файла

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).
Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

2.3.2 Функция SignFile()

Заголовок объявления функции:

CSP_BOOL __attribute__ ((visibility("default"))) VerifyFile(const LPSTR inFileName, const LPSTR DetachedFileName, SignInfo_t*& signInfo);

Входные параметры:

inFileName – имя файла электронной подписи

DetachedFileName – имя файла данных, подпись которого проверяется

Выходные параметры:

signInfo – структура, описывающая информацию о проверенной подписи:

```
typedef struct SignInfo
{
    DWORD signatureIndex;           //номер подписи
    CSP_BOOL isCounterSignature;    //является ли подпись заверяющей
    CSP_BOOL verifyResult;          //математическая корректность
    CSP_BOOL isAdvanced;            //усовершенствованная подпись
    CSP_BOOL verifyCertificateResult;//проверка сертификата
    CSP_BOOL isTimeStampIncluded;  //включен ли штамп времени
    LPWSTR comment;                 //комментарий
    FILETIME signatureTime;         //время подписи
    BYTE* rawSignerCert;            //сертификат
    BYTE* rawSignatureValue;        //сигнатура
    DWORD rawSignerCertSize;        //размер сертификата
    DWORD rawSignatureValueSize;    //размер сигнатуры
} SignInfo_t;
```

Возвращаемое значение:

Если функция завершается успешно, возвращаемое значение отлично от нуля (true).

Если функция завершается ошибкой, возвращаемое значение равно нулю (false).

3 Сообщения системному программисту

Если при выполнении функций криптографической платформы «Litoria Crypto Platform» для iOS возникают ошибочные ситуации, то для получения информации об ошибке необходимо вызвать функцию GetErrors().

Пример вызова функции GetErrors():

```
DWORD ErrorCodeCP;
DWORD* ErrorCodeModule;
DWORD BHGiSErrorCodeSize;
GetErrors(ErrorCodeCP, ErrorCodeModule, BHGiSErrorCodeSize);
fprintf(stderr, "\nError code %d", ErrorCodeCP);
for(int i=0; i<BHGiSErrorCodeSize; i++) fprintf(stderr, "Module error code %d",
ErrorCodeModule[i]);
```

Коды ошибок, возвращаемые криптоплатформой «Litoria Crypto Platform» для iOS, представлены в таблице 3.1.

Таблица 3.1. Коды ошибок

Имя ошибки	Номер ошибки	Текст ошибки
INIT_CREATE_EDS_ERROR	1000	Ошибка инициализации данных для создания ЭП.
INVALID_CERTIFICATE_BLOB	1001	Неверный блок сертификата подписчика.
NOT_FIND_CERT_IN_MY	1002	Сертификат подписчика отсутствует в хранилище «Личные».
NOT_FIND_HASH_ALG	1003	Алгоритм хеширования не определен.
SIGNING_CERTIFICATE_V2_ ERROR	1004	Ошибка формирования атрибута SignerCertReferenseV2.
CREATE_HASH_ERROR	2000	Ошибка вычисления хеш значения.
GET_APPROPRIATE_CP_ERROR	2001	Не удалось найти подходящий криптопровайдер.
GET_HASH_ALG_ID_ERROR	2002	Не удалось получить подходящий алгоритм хеширования.
CREATE_EDS_ERROR	3000	Ошибка создания ЭП.
GET_EXISTING_SIGNS_ERROR	4001	Не удалось получить значения подписей из сообщения.
NULL_HMSG	4002	Неверный дескриптор сообщения.
GET_SIGN_ONLY_ERROR	5001	Не удалось получить значение подписи.
GET_SIGNER_CERT_ERROR	5002	Не удалось получить информацию о сертификате подписавшего лица.
SEND_REQUEST_ERROR	5003	Не удалось получить информацию из сети.
SIGNER_CERT_CHAIN_ERROR	5004	Цепочка сертификата подписавшего лица составлена некорректно.
TSP_CERT_CHAIN_ERROR	5005	Цепочка сертификата tsa оператора составлена некорректно.
OCSP_CERT_CHAIN_ERROR	5006	Цепочка сертификата ocsp оператора составлена некорректно.
CREATE_REVOCATION_ VALUES_ERROR	5007	Ошибка формирования атрибутов RevocationValues и CompleteRevocationRefs.
CREATE_CERTIFICATE_ VALUES_ERROR	5008	Ошибка формирования атрибутов CompleteCertificateRefs и CertificateValues.
CHECK_PROOF_VALIDITY_ ERROR	5009	Ошибка проверки используемых сертификатов на действительность относительно времени внешнего штампа.

Имя ошибки	Номер ошибки	Текст ошибки
SIGNER_CERT_VERIFY_TIME_ ERROR	5010	Один или несколько сертификатов цепочки сертификата подписавшего лица недействителен.
TSP_CERT_VERIFY_TIME_ ERROR	5011	Один или несколько сертификатов цепочки сертификата tsa оператора недействителен.
OCSP_CERT_VERIFY_TIME_ ERROR	5012	Один или несколько сертификатов цепочки сертификата ocsp оператора недействителен.
ADD_UNSIGNED_ATTR_ ERROR	5013	Ошибка добавления неподписанного атрибута в подпись.
GET_RESULT_ERROR	5014	Ошибка получения результат операции.
GET_IN_TIME_STAMP_ERROR	6000	Не удалось получить внутренний штамп времени или он некорректен.
GET_OUT_TIME_STAMP_ ERROR	6001	Не удалось получить внешний штамп времени или он некорректен.
TSP_RESPONSE_STATUS_ ERROR	6002	Статус ответа службы штампов времени некорректен.
CHECK_TSP_SIGNATURE_ ERROR	6003	Полученный штамп времени содержит ошибки.
TSP_CERT_ERROR	6004	Не удалось получить информацию о сертификате tsa оператора.
TSP_NEEDED_OID_ERROR	6005	Сертификат службы штампов времени не содержит необходимый OID установки отметки времени.
VERIFY_TSP_SIGNATURE_ ERROR	6006	Подпись штампа времени математически некорректна.
GET_TIME_IN_TIME_STAMP_ ERROR	6007	Ошибка получения времени создания внутреннего штампа времени.
GET_TIME_OUT_TIME_ STAMP_ERROR	6008	Ошибка получения времени создания внешнего штампа времени.
GET_CRL_ERROR	7000	Необходимый CRL не найден.
GET_LOCAL_CRL_ERROR	7001	Ошибка получения CRL из хранилища.
DOWNLOAD_CRL_ERROR	7002	Ошибка получения CRL из сети.
CERT_CRL_LOCAL_ERROR	7003	Для проверки передан пустой сертификат.
STORE_CA_ERROR	7004	Ошибка доступа к хранилищу промежуточных центров сертификации.
STORE_ROOT_ERROR	7005	Ошибка доступа к хранилищу корневых центров сертификации.
CERT_CRL_REVOKED	7006	Проверяемый сертификат отозван.
GET_CRL_URLS_ERROR	7007	Ошибка получения точек распространения CRL.
CRL_NOT_URLS	7008	Сертификат не содержит ни одной точки распространения CRL.
CRL_CDP_ERROR	7009	Не удалось определить формат CDP.
CRL_TIME_VALIDITY_ERROR	7010	Полученный CRL недействительный по времени.
ADD_CRL_TO_STORE_ERROR	7011	Не удалось установить CRL в хранилище промежуточных центров сертификации.
PROXY_NAME_PASS_ERROR	9000	Необходимые имя пользователя и пароль для прокси-сервера заданы не верно.
VERIFY_TYPE_EDS_ERROR	10000	Ошибка определения типа подписи.
INIT_VERIFY_EDS_ERROR	10001	Ошибка инициализации данных для проверки ЭП.
VERIFY_EDS_VALUE_ERROR	10002	Ошибка проверки математической целостности подписи.

Имя ошибки	Номер ошибки	Текст ошибки
CREATE_STORE_ERROR	11003	Ошибка создания дополнительного хранилища сертификатов.
INVALID_ENCRYPT_ CERTIFICATE_BLOB	12000	Неверный блок сертификата получателя.
INVALID_ENCRYPT_ CERTIFICATE	12001	Сертификат получателя не определен.
FIND_ENCRYPT_OID_ERROR	12002	Не найден подходящий алгоритм шифрования.
FIND_SKEY_ERROR	12003	Не найден закрытый ключ, соответствующий сертификату открытого ключа.
PIN_CODE_ERROR	12004	Набор ключей не существует. Возможно, вы ввели неверный Pin Code.
PROV_NAME_ERROR	13000	Криптопровайдер не указан.
GET_PROV_TYPE_ERROR	13001	Не удалось определить тип криптопровайдера.
CREATE_KEY_CONTAINER_ ERROR	13002	Ошибка создания контейнера ключей.
CREATE_KEY_PAIR_ERROR	13003	Ошибка генерации пары ключей.
GET_PUBLIC_KEY_STRUCT_ LEN_ERROR	13004	Ошибка получения длины структуры под экспортируемый открытый ключ.
GET_PUBLIC_KEY_ERROR	13005	Ошибка экспортирования открытого ключа.
ENCODE_SUBJECT_INFO_ ERROR	13006	Ошибка кодирования информации о владельце сертификата.
SIGN_ADN_ENCODE_CERT_ REQ_ERROR	13007	Возникла ошибка в процессе подписания и кодирования запроса на сертификат.
BASE64_ENCODE_ERROR	13008	Ошибка кодирования информации в base64.
INVALID_CERTIFICATE_ BLOB_FOR_REQ	13009	Неверный блок используемого сертификата.
GET_HPROV_ERROR	13010	Не удалось получить контекст криптопровайдера.
GET_PP_ENUMCONTAINERS_ ERROR	13011	Ошибка при получении размера имени контейнера секретного ключа.
GET_PROV_NAME_ERROR	13012	Ошибка при получении имени контейнера секретного ключа.
GET_CONTAINER_NAME_ ERROR	13013	Имя контейнера ключа не задано и не определено.
SET_PRIVATE_KEY_FOR_ CERT_ERROR	13014	Ошибка создания в сертификате ссылки на закрытый ключ.
ADD_CERT_CONTEXT_TO_ STORE_ERROR	13015	Не удалось установить сертификат в хранилище сертификатов.
BASE64_DECODE_ERROR	13016	Ошибка декодирования информации из base64.
SET_PRIVATE_KEY_ERORR	13018	Ошибка установки сертификата и создания связки открытый - закрытый ключ.
GET_CERT_FROM_STORE_ ERROR	13019	Ошибка получения сертификата из хранилища.
INIT_GET_CERT_LIST_FROM_ STORE_ERROR	13020	Ошибка инициализации данных для получения списка сертификатов из заданного хранилища.
GET_CERT_LIST_FROM_ STORE_ERROR	13021	Ошибка получения списка сертификатов из заданного хранилища.
STRUCT_GET_ENCODE_ ERROR	14000	Ошибка декодирования структуры данных.

Имя ошибки	Номер ошибки	Текст ошибки
STRUCT_BER_DECODE_ERROR	14001	Ошибка кодирования информации в структуру данных.
BH_CERT_TRUST_IS_NOT_ TIME_VALID	16000	Сертификат или один из сертификатов в цепочке не действителен на текущий момент времени.
BH_CERT_TRUST_IS_REVOKED	16001	Сертификат или один из сертификатов в цепочке отозван.
BH_CERT_TRUST_IS_NOT_ SIGNATURE_VALID	16002	Сертификат или один из сертификатов в цепочке искажен (содержит недействительную электронную подпись).
BH_CERT_TRUST_IS_NOT_ VALID_FOR_USAGE	16003	Сертификат или один из сертификатов в цепочке не предназначен для данного использования.
BH_CERT_TRUST_IS_ UNTRUSTED_ROOT	16004	Сертификат или один из сертификатов в цепочке издан недоверенным центром сертификации.
BH_CERT_TRUST_IS_CYCLIC	16005	Один из сертификатов в цепочке издан центром, сертифицированным проверяемым сертификатом (циклическая цепочка).
GET_CSP_NAMES_ERROR	17000	Ошибка получения имен криптопровайдеров в системе.
GET_CSP_PARAMS_ERROR	17001	Ошибка получения параметров криптопровайдера.
GET_CERT_FROM_CONTAINER_ ERROR	17002	Ошибка получения сертификата ключа подписи из контейнера.

Список сокращений

CSP – Cryptographic Service Provider

ГОСТ – ГОсударственный СТандарт

ЭП – Электронная Подпись