УТВЕРЖДЕНО
ИНФК.11485466.4012.017.32 05 ЛУ

Средство криптографической защиты информации "Бикрипт 5.0"

Инструкция программисту

ИНФК.11485466.4012.017.32 05

Листов 206

Москва 2021

ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ 6

1.1. Типы библиотек 6
1.2. Типы ключевой информации 6
1.3. Концепция работы с библиотекой 7
1.4. Особенности работы ДСЧ 8
1.5. Об использовании функций генерации ключей ЭП 8

2. ОПИСАНИЕ БИБЛИОТЕЧНЫХ ФУНКЦИЙ 8

1 Функции инициализации и деинициализации 8

1.1. Инициализация библиотеки 8
1.2. Деинициализация библиотеки 12
1.3. Создание функции вывода прогресса исполнения 13
1.4. Получение информации об используемой библиотеке 14
1.5. Получение номера ТМ-идентификатора, присоединенного к считывателю 15
1.6. Установка параметров библиотеки 16
1.7. Получение параметров библиотеки 17
1.8. Инициализация программного датчика случайных чисел (ПДСЧ) 17
1.9. Инициализация программного датчика случайных чисел (ПДСЧ) 18
1.10. Сохранение ключа ПДСЧ в файл или на ТМ 19
1.11. Загрузка дополнительных библиотек 20
1.12. Универсальная функция инициализации 21

2 Функции вычисления значений хеш-функции 24

2.1. Инициализация вычислений значений хеш-функции 24
2.2. Вычисление значения хеш-функции для части блока данных 24
2.3. Завершение вычисления значения хеш-функции для блока данных 25
2.4. Установка значения хеш-функции 26
2.5. Создание копии дескриптора хеш-функции 27
2.6. Закрытие дескриптора хеш-функции 28

3 Функции формирования и загрузки ключей ЭП 28

3.1. Генерация мастер-ключа 28
3.2. Установка мастер-ключа 29
3.3. Перешифрование мастер-ключа на новом Главном ключе 30
3.4. Генерация ключей ЭП с "привязкой" к ТМ 31
3.5. Генерация ключей ЭП в файл с паролем 33
3.6. Генерация ключей ЭП с маской - мастер-ключом 35
3.7. Генерация ключей ЭП в файл с маской - мастер-ключом 37
3.8. Загрузка закрытого ключа ЭП с паролем или "привязкой" к ТМ 39
3.9. Загрузка из файла закрытого ключа ЭП с паролем 41
3.10. Загрузка закрытого ключа ЭП с мастер-ключом 42
3.11. Загрузка из файла закрытого ключа ЭП с мастер-ключом 44
3.12. Получение идентификатора закрытого ключа ЭП 46
3.13. Выгрузка закрытого ключа ЭП 47
3.14. Генерация открытого ключа 47
3.15. Генерация главного ключа 48
3.16. Копирование главного ключа в ТМ 50
3.17. Расширенная генерация ключей ЭП с паролем или “привязкой” к ТМ 51
3.18. Расширенная генерация ключей ЭП с маской - мастер-ключом 53
3.19. Замена закрытого ключа ЭП с паролем или “привязкой” к ТМ 55
3.20. Создание временного закрытого и открытого ключа ЭП 57

4 Функции работы со справочником открытых ключей ЭП 58

4.1. Открытие справочника открытых ключей 58
4.2. Расширенное открытие справочника открытых ключей 59

4.3. Открытие справочника из памяти 60
4.4. Проверка наличия открытого ключа ЭП в справочнике 61
4.5. Добавление/замена открытого ключа ЭП в справочник(е) 62
4.6. Удаление открытого ключа ЭП из справочника 63
4.7. Сохранение справочника 63
4.8. Выбор первого ключа из справочника открытых ключей 64
4.9. Выбор следующего ключа из справочника открытых ключей 65
4.10. Закрытие справочника 66
4.11. Запись в ТМ-идентификатор 32-байтного значения 67
4.12. Чтение из ТМ-идентификатора 32-байтного значения 68
4.13. Чтение трех страниц из ТМ-идентификатора 69
4.14. Установка в контекст открытого ключа 69
4.15. Загрузка открытого ключа из области памяти 70
4.16. Получение идентификатора открытого ключа 71
4.17. Экспортирование открытого ключа 72
4.18. Деинициализация открытого ключа 72
4.19. Получение информации о средствах ЭП сертификата X.509 73

5 Функции формирования и проверки ЭП 74
5.1. Формирование ЭП для блока памяти 74
5.2. Проверка ЭП для блока памяти 76
5.3. Формирование ЭП для значения хеш-функции 77
5.4. Проверка ЭП для значения хеш-функции 78
5.5. Формирование ЭП для содержимого файла 79
5.6. Формирование ЭП для содержимого файла, открытого приложением 80
5.7. Проверка ЭП для содержимого файла 82
5.8. Помещение структуры данных подписи в конец буфера 83
5.9. Помещение структуры данных подписи в конец файла 84
5.10. Получение структуры данных подписи из буфера 85
5.11. Получение структуры данных подписи из файла 87

6 Функции ДСЧ 88
6.1. Генерация случайной последовательности 88

7 Функции верхнего уровня 89
7.1. Служебные функции 89
7.1.1. Получение следующего индекса цепочки сертификатов 89
7.1.2. Получение корневого сертификата в цепочке сертификатов 89
7.1.3. Получение имени файла сертификата в хранилище 90
7.1.4. Получение сертификата из хранилища 91
7.1.5. Получение статуса сертификата из хранилища 91
7.1.6. Получение длины цепочки сертификатов в хранилище 92
7.1.7. Получение количества сертификатов и списка отозванных сертификатов в хранилище 93
7.1.8. Создание контекста времени 93
7.1.9. Освобождение контекста времени 94
7.1.10. Установка текущего времени для контекста времени 95
7.1.11. Установка заданного времени в контекст 95
7.1.12. Получение времени из контекста 96
7.1.13. Копирование контекста времени 97
7.1.14. Политика сертификата и класс средства ЭП 98
7.1.15. Удаление ЭП из CMS 99
7.1.16. Получение информации об ЭП из CMS в буфере 100
7.1.17. Получение информации об ЭП из CMS в файле 102
7.1.18. Получение времени ЭП и выделенного имени подписанта CMS в буфере 103
7.1.19. Получение времени ЭП и выделенного имени подписанта CMS в файле 105
7.1.20. Получение времени ЭП и выделенного имени создателя списка отозванных сертификатов 106
7.1.21. Получение времени ЭП и выделенного имени владельца из сертификата в буфере 108

7.1.22. Получение времени ЭП и выделенного имени владельца из сертификата в хранилище 109
7.1.23. Завершение работы 110
7.1.24. Вызов конфигуратора 111
7.2. Функции для реализации электронной подписи в соответствии со стандартом X.509 111
7.2.1. Создание ключевой пары на основе выделенного имени и атрибутов, кодированных в формате ASN.1 111
7.2.2. Создание ключевой пары на основе данных, идентифицирующих владельца сертификата 114
7.2.3. Инициализация контекста библиотеки 118
7.2.4. Добавление сертификата в контекст библиотеки 119
7.2.5. Добавление списка отозванных сертификатов в контекст библиотеки 120
7.2.6. Добавление сертификатов и списков отозванных сертификатов из файлов в контекст библиотеки 122
7.2.7. Инициализация датчика ПДСЧ 124
7.2.8. Создание копии контекста библиотеки 126
7.2.9. Освобождение контекста библиотеки 127
7.2.10. Получение количества ключевых пар в контексте библиотеки 127
7.2.11. Получение информации о ключевой паре в контексте библиотеки 128
7.2.12. Получение информации о носителе ключа ПДСЧ в контексте библиотеки 129
7.2.13. Поиск ключевой пары по сертификату в контексте библиотеки 130
7.2.14. Загрузка ключевой пары в контексте библиотеки 131
7.2.15. Освобождение ресурсов ключевой пары 133
7.2.16. Инициализация контекста вычисления значений хеш-функции 134
7.2.17. Задание параметров контекста вычисления значений хеш-функции 134
7.2.18. Получение параметров контекста вычисления значений хеш-функции 135
7.2.19. Задание параметров контекста вычисления значений хеш-функции, согласованных с ключевой парой 136
7.2.20. Добавление порции данных в контекст вычисления значений хеш-функции 136
7.2.21. Получение значения хеш-функции для буфера памяти 137
7.2.22. Получение значения в контексте вычисления значений хеш-функции 138
7.2.23. Закрыть контекст вычисления значений хеш-функции 138
7.2.24. Инициализация контекста операции формирования ЭП 139
7.2.25. Добавление значения хеш-функции в контекст операции формирования ЭП 140
7.2.26. Накопление данных в контексте операции формирования ЭП 141
7.2.27. Добавление в контекст операции формирования ЭП существующей CMS 141
7.2.28. Формирование ЭП в контексте операции формирования ЭП 142
7.2.29. Закрытие контекста операции формирования ЭП 144
7.2.30. Электронная подпись области памяти 144
7.2.31. Электронная подпись файла 146
7.2.32. Электронная подпись значения хеш-функции 149
7.2.33. Инициализация контекста операции проверки ЭП 151
7.2.34. Добавление значения хеш-функции в контекст операции проверки ЭП 151
7.2.35. Накопление данных для проверки отсоединенной ЭП в контексте операции проверки ЭП 152
7.2.36. Добавление в контекст операции проверки ЭП существующей CMS 153
7.2.37. Проверка ЭП и получение результатов в контексте операции проверки ЭП 153
7.2.38. Закрытие контекста операции проверки ЭП 155
7.2.39. Проверка ЭП области памяти 156
7.2.40. Проверка ЭП области памяти 159
7.2.41. Проверка ЭП файла 161
7.2.42. Проверка ЭП файла 164
7.2.43. Проверка ЭП значения хеш-функции 166
7.2.44. Проверка ЭП значения хеш-функции 169
7.2.45. Проверка ЭП сертификата из хранилища 171
7.2.46. Проверка ЭП сертификата из буфера по хранилищу 173
7.2.47. Проверка ЭП запроса на сертификат 175
7.2.48. Проверка ЭП списка отозванных сертификатов по хранилищу 176
7.2.49. Освобождение контекста пути доступа к ключу 178

7.3. Функции для реализации шифрования в соответствии со стандартом X.509 179
7.3.1. Инициализация контекста операции зашифрования 179
7.3.2. Накопление данных в контексте операции зашифрования 179
7.3.3. Выполнение шифрования блока данных в контексте операции зашифрования 180
7.3.4. Закрытие контекста операции зашифрования 182
7.3.5. Зашифрование области памяти 182
7.3.6. Инициализация контекста операции расшифрования 184
7.3.7. Добавление в контекст операции расшифрования существующей CMS 184
7.3.8. Выполнение расшифрования блока данных в контексте операции расшифрования 185
7.3.9. Закрытие контекста операции расшифрования 186
7.3.10. Расшифрование области памяти 187
7.4. Функции центра сертификации 189
7.4.1. Создание подписанного CMS включающего PKCS10 запрос 189
7.4.2. Создание подписанного ключом администратора ЦС сертификата на основе запроса 190
7.4.3. Создание ключевой пары администратора ЦС на основе ключа в формате БиКрипт 191
7.4.4. Создание корневого сертификата 192
7.4.5. Создание подписанного ключом администратора ЦС списка отозванных сертификатов 193

8 Используемые дескрипторы 194

9 Используемые константы 194

ПРИЛОЖЕНИЕ №1
195

ПРИЛОЖЕНИЕ №2
199

ПРИЛОЖЕНИЕ №3
204

ПРИЛОЖЕНИЕ №4
206

1. Введение

В данной документации содержится описание библиотек СКЗИ "Бикрипт 5.0".

СКЗИ "Бикрипт 5.0" является средством защиты конфиденциальной информации и реализует криптографические процедуры: генерация ключей электронной подписи (ЭП), формирование ЭП, проверка ЭП, вычисление значения хеш-функции и шифрование.

В состав СКЗИ входят программные модули:

- реализующие контроль целостности hashctrl;
- реализующие программный датчик случайных чисел Grn.dll и Grn64.dll (варианты исполнений 1, 2, 9, 10), libbiogrn.dylib и libbiogrn_64.dylib (варианты исполнений 15, 16);
- реализующие генерацию ключей электронной подписи (ЭП), формирование ЭП, проверку ЭП, вычисление значения хеш-функции библиотеки bicr5exp.dll и bicr5exp_64.dll (варианты исполнений 1, 9), libbicr5exp.dylib и libbicr5exp.dylib (вариант исполнения 15);
- реализующие генерацию ключей электронной подписи (ЭП), формирование ЭП, проверку ЭП, вычисление значения хеш-функции и шифрования библиотеки bicr5.dll и bicr5_64.dll (варианты исполнений 2, 10), libbicr5.so и libbicr5_64.so (варианты исполнений 3, 4, 5, 11), libbicr5.dylib и libbicr5_64.dylib (вариант исполнения 16);
- реализующие функции электронной подписи в соответствии со стандартами X.509 и PKCS#7 библиотеки cryptox509.dll и cryptox509_64.dll (варианты исполнений 1, 2, 9, 10), libcryptox509.so и libcryptox509_64.so (варианты исполнений 3, 4, 5, 11), libcms80exp.dylib и libcms80exp_64.dylib (вариант исполнения 15), libcms80.dylib и libcms80_64.dylib (варианты исполнений 16). Использование этих функций (см. раздел 7) при разработке прикладного ПО с применением СКЗИ "Бикрипт 5.0" возможно без проведения дополнительных тематических исследований, при использовании в прикладном ПО вызовов прочих функций необходимо производить разработку отдельного варианта исполнения СКЗИ или нового СКЗИ.

Программные библиотеки предназначены для использования в среде разработки языка программирования C.

Библиотеки Grn.dll (Grn64.dll), libbiogrn.dylib (libbiogrn_64.dylib) для корректного функционирования должны находиться в текущем каталоге приложения, использующего библиотеки СКЗИ.

1.1. Типы библиотек

Программные библиотеки реализованы для семейств операционных систем Unix, Windows и macOS Особенности функционирования библиотеки под платформами Unix и Windows указаны в Приложении №4.

В комплекте разработчика библиотеки bicr5.dll и libbicr5 могут быть представлены в стандартном и специальном (sign_only) варианте. Стандартная и специальная (sign_only) версии библиотек различаются списком доступных функций. Список функций для стандартной и специальной (sign_only) версии библиотеки приведен в Приложении №2. В стандартной версии названия библиотечных файлов bicr5.dll, bicr5_64.dll, libbicr5.dylib и libbicr5_64.dylib. В специальной версии названия библиотечных файлов bicr5exp.dll, bicr5exp_64.dll, libbicr5exp.dylib и libbicr5exp_64.dylib.

1.2. Типы ключевой информации

В СКЗИ "Бикрипт 5.0" используются следующие типы ключевой информации:

- Главный ключ (ГК).

  ГК представляет собой 32-байтную случайную последовательность, используется при организации персонального доступа к ключевой информации. Хранится на отчуждаемом ключевом носителе (НГМД, ТМ или Flash Memory).

- Закрытый ключ ЭП.

  Используется для формирования ЭП данных, представляет собой 32-байтную случайную последовательность. Закрытый ключ ЭП может использоваться в двух форматах:

  - Однокомпонентный (закрытый ключ полностью хранится в носителе).
  - Двухкомпонентный (закрытый ключ разложен на две 32-байтные компоненты, одна из которых хранится на отчуждаемом носителе, а вторая (мастер-ключ ЭП), в зашифрованном на ГК виде, на НЖМД ПЭВМ).

- Открытый ключ ЭП.

  Формируется из закрытого ключа ЭП в соответствии с ГОСТ Р 34.10-2012, хранится (как свой, так и полученные) с имитозащитой (защитой от внедрения ложных данных).

- Ключ программного датчика случайных чисел.

  Ключ программного датчика случайных чисел используется для создания ключевой информации и вырабатывается на рабочем месте с использованием программного ДСЧ. Ключ программного датчика случайных чисел хранится на отчуждаемом ключевом носителе.

- База открытых ключей ЭП.

  База открытых ключей ЭП хранит открытые ключи ЭП пользователей и может использоваться для проверки ЭП данных, как и отдельный открытый ключ.

1.3. Концепция работы с библиотекой

Общая схема работы с библиотекой выглядит следующим образом:

1. Загрузка библиотек
2. Инициализация библиотеки
3. Инициализация датчика случайных чисел;
4. Выполнение необходимых криптографических операций;
5. Деинициализация библиотеки.

Загрузка библиотек должна быть выполнена до инициализации библиотеки. Данная процедура производится при помощи функции cr_load_bicr_dll или при помощи универсальной функции инициализации cr_init_bicr4.

Процедура инициализации библиотеки инициализирует внутренние структуры библиотеки, определяет наличие и работоспособность различных системных компонент и выделяет необходимые ресурсы.

Результатом инициализации библиотеки является возвращенный дескриптор контекста СКЗИ "Бикрипт 5.0". Для многопоточных приложений с целью увеличения производительности может быть проинициализировано несколько контекстов СКЗИ "Бикрипт 5.0", число контекстов соответствует числу одновременно работающих потоков.

Контекст инициализации СКЗИ "Бикрипт 5.0" возвращается в любом случае (кроме случая, когда возвращаемое значение дескриптора установлено в NULL). Условно библиотека может быть проинициализирована в 3-х режимах работы:

- В режиме позволяющем проводить все криптографические операции (следует инициализировать библиотеку Главным ключом и проинициализировать ДСЧ);
- В режиме позволяющем проводить вычисление хеш-функции, формирование и проверку ЭП, генерацию открытого и закрытого ключей ЭП (достаточно инициализировать библиотеку и ДСЧ);
- В режиме позволяющем проводить вычисление хеш-функции, проверку ЭП (достаточно осуществить только инициализацию библиотеки без инициализации ДСЧ и ввода ГК).

Режим работы библиотеки определяется значением параметра init_mode по окончании процесса инициализации библиотеки (и ДСЧ при необходимости). Значение параметра init_mode определяется совокупностью значений следующих битовых флагов:

- Флаг инициализации Главного ключа (cGKUZ);
- Флаг инициализации ДСЧ (cDSCH);
- Флаг инициализации драйвера работы с ТМ (cTMDRV).
- Флаг индикации присутствия в компьютере VPN-Key (cVPNKEY)[1].

Совокупность значений флагов непосредственно определяет режим работы библиотеки, а также возможность использования внешних устройств для хранения закрытых ключей ЭП. Подробное описание битовых флагов и доступных функций приведено в описании библиотечной функции cr_init.

Любая библиотечная функция при неуспешном выполнении возвращает код ошибки. Полный список кодов ошибок приведен в Приложении №1.

1.4. Особенности работы ДСЧ

Библиотека поддерживает работу с программным датчиком случайных чисел (ПДСЧ). ПДСЧ инициализируется пользователем движениями мыши.

Инициализация ПДСЧ производится один раз при первом запуске СКЗИ. В процессе инициализации на отчуждаемый носитель записывается ключ ПДСЧ, который в дальнейшем может использоваться для инициализации ПДСЧ без дополнительных действий пользователя.

Инициализация ДСЧ производится после инициализации библиотеки вызовом функции cr_init_prnd.

1.5. Об использовании функций генерации ключей ЭП

Функции генерации и загрузки ключей ЭП cr_read_skey, cr_read_skey_mk, cr_gen_keypair и cr_gen_keypair_mk следует использовать для создания ключей при работе с отчуждаемым носителем (USB-флеш диск или дискета).

Функции cr_gen_elgkey, cr_gen_elgkey_ext, cr_gen_elgkey_mk, cr_gen_elgkey_ext_mk, cr_load_elgkey и cr_load_elgkey_mk следует использовать **только** для работы с ТМ – идентификатором (**параметр secrkey=NULL**).

2. Описание библиотечных функций

1 Функции инициализации и деинициализации

1.1. Инициализация библиотеки

```
#ifdef SIGN_ONLY
int DECL cr_init(int tm_flag,
                 int *init_mode,
                 H_INIT *init_handle);
#else
int DECL cr_init(int tm_flag,
                 const char *gk,
                 const char *uz,
                 const char *psw,
                 void *tm_number,
                 int *tmn_blen,
                 int *init_mode,
                 H_INIT *init_handle );
```

[1] Не используется при работе в ОС macOS.

Назначение:

Функция определяет наличие и работоспособность различных системных компонент, выделяет необходимые ресурсы и осуществляет загрузку с носителей первичного ключа хранения (главного ключа), узла замены, при их наличии.

Параметры:

tm_flag	in	0 – если не требуется использовать специализированные носители ключевой информации 1 - если в дальнейшем предполагается работа с ТМ-идентификатором
gk, uz	in	путь к файлам с главным ключом и узлом замены; если носителем главного ключа и узла замены является ТМ, данные параметры должны принимать значение NULL. Если для работы использование главного ключа и узла замены не требуется, то данный параметр можно установить в "\0" или в "". В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
psw	in	пароль, на котором зашифрован главный ключ при хранении; если главный ключ хранится в открытом виде - данный параметр должен принимать значение "\0". В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
tm_number	out	указатель на буфер, в который возвращается номер ТМ. Если носителем ключевой информации является файл (дискета), данный параметр игнорируется В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе – реальную возвращенную длину буфера tm_number, если носителем ключевой информации является файл (дискета), данный параметр игнорируется. В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
init_mode	out	возвращаемое значение, которое указывает на наличие драйверов, используемых в процессе работы библиотеки, и результат ввода главного ключа

init_mode = cGKUZ | cDSCH | cTMDRV | cVPNKEY

Значения соответствующих флажков в битовом представлении таковы:

cGKUZ	00000001	главный ключ корректно введен
cDSCH	00000010	ДСЧ обнаружен
cTMDRV	00000100	драйвер работы с ТМ обнаружен
cVPNKEY[2]	00010000	токен VPN-Key обнаружен

init_handle	out	возвращаемый дескриптор H_INIT, используемый другими функциями библиотеки, характеризующий контекст библиотеки. В случае неустранимой ошибки устанавливается в NULL. В случае простой ошибки возвращается ненулевое значение и с библиотекой можно работать.

Возвращаемое значение:

Функция возвращает код ошибки, который, в случае ненулевого возврата init_handle, должен интерпретироваться как предупреждение и с библиотекой можно работать:

код ошибки	Описание
ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

[2] Не используется при работе в ОС macOS.

ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_OPEN_FILE	Ошибка открытия файла
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция должна быть вызвана в начале сессии работы с библиотекой для данного приложения. Эта функция должна быть вызвана первой из функций библиотеки. После завершения сессии работы с библиотекой в данном приложении должна быть вызвана функция cr_uninit().
При вызове функции с параметром tm_flag=1 проверяется возможность работы со считывателем touch memory (таблеткой, ТМ). Для работы с таблеткой в ОС Windows 7, Windows 8, Windows 8.1, Windows 10, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008 (разрядность x86/x64) и Windows Server 2008 R2, Windows Server 2012, Windows Server 2012 R2, Windows Server 2016, Windows Server 2019 (разрядность x64) необходимо загрузить драйвер TMDRV.SYS версии 2.03 и выше. Драйверы для работы с ТМ через плату Аккорд доступны по ссылке http://www.infocrypt.ru/tmdrvsys.zip, для USB-считывателя - ИнфоКрипт (infocrypt.ru) Для работы с таблеткой в ОС macOS необходимо установить стандартную библиотеку libusb.
При вызове функции с параметром tm_flag=2 проверяется возможность работы с VPN-Key. Не используется при работе в ОС macOS. Для успешной инициализации функции с этим параметром на ПК должно быть установлено ПО "ФПСУ/IP-Клиент" версии 3.1.2 или выше.
Функция проверяет наличие в памяти драйверов для работы с ТМ и через параметр init_mode возвращает информацию об их наличии, а также результат ввода главного ключа.

init_mode = cGKUZ | cDSCH | cTMDRV | cVPNKEY

где параметры имеют следующие значения в битовом представлении:

cGKUZ	00000001	главный ключ корректно введен	код ошибки ERR_NOT_USED_GKUZ В версии без шифрования (SIGN_ONLY) эта ошибка не возвращается
cDSCH	00000010	ключ ПДСЧ загружен	код ошибки ERR_NOT_USED_DSCH
cTMDRV	00000100	драйвер работы с ТМ обнаружен	код ошибки ERR_NOT_USED_TMDRV
cVPNKEY[3]	00010000	VPN-Key обнаружен и готов к работе	код ошибки ERR_NOT_USED_VPNKEY

В зависимости от значения параметра init_mode при дальнейшей работе библиотеки доступны те или иные функции. Если функция недоступна, выдается код ошибки ERR_NOT_USED_TMDRV или, соответственно, ERR_NOT_USED_DSCH , ERR_NOT_USED_GKUZ.

	TMDRV/ VPNKEY нет	ДСЧ нет	ГК не введен	Версия без шифрования SIGN_ONLY
cr_init()	+	+	+	+

[3] Не используется при работе в ОС macOS.

cr_init_bicr4()	+	+	+	+
cr_uninit()	+	+	+	+
cr_init_random()	+	+	+	+
cr_set_procent_callback()	+	+	+	+
cr_get_version_info()	+	+	+	+
cr_set_param()	+	+	+	+
cr_get_param()	+	+	+	+
cr_get_tm_number()	+	+	+	+
cr_hash_open()	+	+	+	+
cr_hash_calc()	+	+	+	+
cr_hash_return()	+	+	+	+
cr_hash_set()	+	+	+	+
cr_hash_dup()	+	+	+	+
cr_hash_close()	+	+	+	+
cr_gen_masterkey()	+	-	+	-
cr_install_masterkey()	+	-	-	-
cr_change_masterkey()	+	+	-	-
cr_gen_onetime_keypair()	+	-	+	+
cr_gen_elgkey()	+/- (доступна генерация закрытого ключа только в файл)	-	+	+
cr_gen_elgkey_ext()	+/- (доступна генерация закрытого ключа только в файл)	-	+	+
cr_gen_keypair()	+	-	+	+
cr_gen_elgkey_mk()	+/- (доступна генерация закрытого ключа только в файл)	-	-	-
cr_gen_elgkey_mk_ext()	+/- (доступна генерация закрытого ключа только в файл)	-	-	-
cr_gen_keypair_mk()	+	-	+	+
cr_load_elgkey()	+/- (ввод из файла/ с ТМ)	+/- (ключ вводится/возвращается только номер ТМ)	+	+
cr_read_skey()	+	+	+	+
cr_change_elgkey()	+/- (ввод из файла/ с ТМ)	+/- (ключ вводится/возвращается только номер ТМ)	+	+

cr_load_elgkey_mk()	+/- (ввод из файла/ с ТМ)	+/- (ключ вводится/возвращается только номер ТМ)	+/- (ключ вводится/возвращается только номер ТМ)	-
cr_read_skey_mk()	+	+	+	+
cr_elgkey_getid()	+	+	+	+
cr_elgkey_close()	+	+	+	+
cr_gen_pubkey()	+	+	+	+
cr_gen_gk()	+/- (ввод из файла/ с ТМ)	-	+	-
cr_gk_copy_tm()	-	+	+	-
cr_gk_read_tm()	-	+	+	-
cr_pkbase_open()	+	+	+	+
cr_pkbase_open_membuf()	+	+	+	+
cr_pkbase_find()	+	+	+	+
cr_pkbase_add()	+	+	+	+
cr_pkbase_remove()	+	+	+	+
cr_pkbase_save()	+	+	+	+
cr_pkbase_findfirst()	+	+	+	+
cr_pkbase_findnext()	+	+	+	+
cr_pkbase_close()	+	+	+	+
cr_write_tm	-	+	+	+
cr_read_tm	-	+	+	+
cr_read_three_tm_page()	-	+	+	+
cr_pkey_put()	+	+	+	+
cr_pkey_load()	+	+	+	+
cr_pkey_getid()	+	+	+	+
cr_pkey_getinfo()	+	+	+	+
cr_pkey_close()	+	+	+	+
cr_sign_buf()	+	-	+	+
cr_check_buf()	+	+	+	+
cr_sign_hash()	+	-	+	+
cr_check_hash()	+	+	+	+
cr_sign_file()	+	-	+	+
cr_check_file()	+	+	+	+
cr_buf_put_sign_struct()	+	+	+	+
cr_file_put_sign_struct()	+	+	+	+
cr_buf_get_sign_struct()	+	+	+	+
cr_file_get_sign_struct()	+	+	+	+
cr_get32_random()	+	-	+	+

"+" - функция доступна,
"-" - функция недоступна.

- Примечание 1. Номер ТМ, если ТМ - идентификатор приложен к съемнику, или серийный номер VPN-Key, если USB-токен установлен в USB-порт, возвращаются независимо от того, успешно введен главный ключ или нет.

- Примечание 2. В версии без шифрования (SIGN_ONLY) функция, помеченная значком "Минус", не просто недоступна для вызова, а ее физически нет в библиотеке.

1.2. Деинициализация библиотеки

int DECL **cr_uninit*(* H_INIT** *init_handle* ***);***

Назначение:

Функция освобождает дескриптор H_INIT и соответствующие ресурсы.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти

Описание функции:

Функция запускается в конце сессии работы с библиотекой для данного приложения.
Она освобождает дескриптор H_INIT и ресурсы, распределенные функцией cr_init(), при этом ключевая информация в памяти компьютера затирается.

1.3. Создание функции вывода прогресса исполнения

int DECL **cr_set_procent_callback (H_INIT** *init_handle***,**
int(***ProcentFunc*)(**void***,**long**,**long**),**
void ****ProcentData*);**

Назначение:

Данная функция работает в паре с функциями cr_sign_file(), cr_check_file(),cr_hash_calc() и предназначена для вывода прогресса исполнения данных функций в виде процента готовности.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
ProcentFunc	in	указатель на функцию вывода процента, если задать NULL, то отменяет функцию вывода процента готовности
ProcentData	in	указатель на данные для функции ProcentFunc, эти данные будут передаваться как первый параметр при вызове ProcentFunc

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Функция передает библиотеке шифрования и подписи указатель на функцию ProcentFunc (функцию вывода прогресса исполнения функций). В дальнейшем при вызове любой из функций cr_sign_file(), cr_check_file(), cr_hash_calc() будет автоматически вызвана функция ProcentFunc() с параметрами:

(void *) – первый параметр будет равен Procent_Data (определяемая пользователем структура, можно задавать как NULL)
(long) – второй параметр, равен степени готовности файла (в байтах)
(long) – третий параметр, равен общей длине файла (в байтах)

- Примечание 1. Если в процессе использования функции cr_set_procent_callback установить ProcentFunc() в NULL, то ProcentFunc() более не будет вызываться из функций cr_sign_file(), cr_check_file(), cr_hash_calc().

- Примечание 2. Если ProcentFunc() вернет значение ERR_STOP (28) то соответствующее вычисление функции cr_sign_file(), cr_check_file() или cr_hash_calc() остановится и также вернет ошибку ERR_STOP (28). Эту особенность функции ProcentFunc() можно использовать, если нужно прервать исполнение функции cr_sign_file(), cr_check_file() или cr_hash_calc(), не дожидаясь окончания ее работы.

1.4. Получение информации об используемой библиотеке

int DECL **cr_get_version_info**(**char** *_info_str_,
int *_str_blen_);

Назначение:

Функция позволяет получить информацию о текущей версии библиотеки "Бикрипт 5.0"

Параметры:

info_str	out	указатель на буфер, в который возвращается информация о библиотеке подписи, размер не более 80 байт
str_blen	in/out	на входе содержит максимальную длину буфера info_str, на выходе – реальную возвращенную длину буфера info_str

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция возвращает информацию о текущей версии библиотеки, включая дату создания библиотеки.

- Эллиптическая версия содержит в возвращаемом буфере подстроку Ellip
- Версия SIGN_ONLY содержит в возвращаемом буфере подстроку SIGN
- Стандартная версия библиотеки (bicr5.dll, libbicr5.dylib - дополнительно реализованы функции генерации ключей) содержит в возвращаемом буфере подстроку ADM

1.5. Получение номера ТМ-идентификатора, присоединенного к считывателю

int DECL **cr_get_tm_number (H_INIT** *init_handle***,**
void *****tm_number*,**
int *****tmn_blen***);**

Назначение:

Функция читает присоединенный к считывателю ТМ-идентификатор и возвращает его номер.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
tm_number	out	указатель на буфер, в который возвращается номер ТМ
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе – реальную возвращенную длину буфера tm_number

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора

Описание функции:

Функция считывает номер ТМ. Если в момент вызова функции ТМ устройство не готово (не приложено к считывателю), возвращается соответствующий код ошибки. Номер ТМ устройства может быть использован для идентификации ключевого носителя.

Номер ТМ, возвращаемый данной функцией, имеет размер 10 байт: XXYYYYYYYY.
Первый байт соответствует типу ТМ, остальные - серийный номер производителя и его CRC.
В настоящее время обычно используются ТМ следующих типов:

Dallas1992 (объем памяти 128 байт) - тип 08.
Dallas1993 (объем памяти 512 байт) - тип 06.
Dallas1996 (объем памяти 8192 байт)- тип 0C. Поддерживается также тип 8C.

1.6. Установка параметров библиотеки

int DECL **cr_set_param (BICR_HANDLE** *handle***,**
int *option***,**
int *value****);***

Назначение:

Функция устанавливает параметры криптографических алгоритмов для проинициализированных ранее дескрипторов H_INIT, H_USER, H_PKEY.

Параметры:

handle	in	дескриптор H_INIT, возвращенный функцией cr_init() или дескриптор H_USER, возвращенный функцией cr_read_skey() или дескриптор H_PKEY, возвращенный функцией cr_pkey_load()
option	in	номер опции, принимает значение 1
value	in	значение опции option для опции option=1: установка параметров криптографических алгоритмов – *см. Приложение №3*

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция устанавливает параметры работы криптографических алгоритмов. Функция может устанавливать параметры эллиптической кривой для дескриптора H_INIT, как указано в таблице. Таким образом, после установки option=1 value=65 последующий вызов функции создания ключа, например, cr_gen_keypair() создаст ключ с параметром id-GostR3410-2001-CryptoPro-A-ParamSet из RFC 4357.

1.7. Получение параметров библиотеки

int DECL **cr_get_param (BICR_HANDLE** *handle***,**
int *option***,**
int ****value**);***

Назначение:

Функция возвращает параметры криптографических алгоритмов для проинициализированных ранее дескрипторов H_INIT, H_USER, H_PKEY.

Параметры:

handle	in	дескриптор H_INIT, возвращенный функцией cr_init() или дескриптор H_USER, возвращенный функцией cr_read_skey() или дескриптор H_PKEY, возвращенный функцией cr_pkey_load()
option	in	номер опции, принимает значение 1
value	out	значение опции option для опции 1: получение текущих параметров*, см. Приложение №3*

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее

Описание функции:

Функция возвращает параметры криптографических алгоритмов.
Функция также может возвращать параметры эллиптической кривой для дескрипторов H_INIT, H_USER, H_PKEY как указано в таблице. Таким образом, можно загрузить ключ, например, при помощи функций cr_read_skey() или cr_pkey_find() и проверить его параметры эллиптической кривой по RFC 4357.

1.8. Создание ключа программного датчика случайных чисел (ПДСЧ)

int DECL **cr_init_random (HWND** *hWnd***);**

Назначение:

Создает ключ программного датчика случайных чисел с помощью движений мыши.

Параметры:

В ОС macOS функция используется без параметров.

hWnd	in	только в ОС Windows. дескриптор родительского окна, если родительского окна нет, может принимать значение NULL.

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)

Описание функции:

Создание ключа программного датчика случайных чисел.
Функция использует библиотеку grn.dll или grn64.dll (в вариантах исполнения 1, 2), libbiogrn.dylib или libbiogrn_64.dylib (в вариантах исполнения 15, 16).
Вызов этой функции выводит на экран окно, в котором пользователь должен перемещать указатель мыши в рабочем окне на экране компьютера, многократно меняя направление движения по горизонтальной оси. Если инициализация ПДСЧ прошла неудачно, то из криптографических функций будут доступны только функции по проверке ЭП.

1.9. Инициализация программного датчика случайных чисел

int DECL **cr_init_prnd (H_INIT** *init_handle***,**
char * *prnd_filename***,**
int *flag_init_grn***);**

Назначение:

Функция инициализирует программный датчик случайных чисел.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init()
prnd_filename	in	имя файла, где хранится ключ ПДСЧ, или NULL, если ключ хранится в ТМ-идентификаторе файл должен располагаться на отчуждаемом носителе (дискете или флешке)
flag_init_grn	in	В случае ошибки при чтении файла с ключом ПДСЧ: если flag_init_grn = 1, то для инициализации ПДСЧ будет автоматически использован ПКДСЧ из библиотеки grn.dll, grn64.dll, libbiogrn.dylib или libbiogrn_64.dylib если flag_init_grn = 0, то библиотеки grn.dll, grn64.dll, libbiogrn.dylib и libbiogrn_64.dylib не будут использоваться и будет выдана ошибка

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_DSCH	Ошибка ПДСЧ
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYP	Ошибка шифрования
ERR_READ_FILE	Ошибка чтения файла
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция инициализирует программный датчик случайных чисел. При этом происходит чтение файла или ТМ, прочитанные данные модифицируются и записываются обратно. Файл должен располагаться на отчуждаемом носителе (дискете или флешке), доступном для записи.

Если чтение файла или ТМ произошло с ошибкой (или файл не существует) и если flag_init_grn=1, то будет вызван ПКДСЧ из библиотеки grn.dll или grn64.dll (в вариантах исполнения 1, 2), libbiogrn.dylib или libbiogrn_64.dylib (в вариантах исполнения 15, 16), создающий ключ ПДСЧ, который может быть сохранен при помощи функции cr_write_prnd в файл или на ТМ для дальнейшего использования.

Если имя файла задать пустым в виде "\0", в этом случае файл будет создан в директории пользователя user (домашняя директория пользователя) с именем prnd.db3.

1.10. Сохранение ключа ПДСЧ в файл или на ТМ

int DECL **cr_write_prnd (H_INIT** *init_handle***,**
char ***_prnd_filename_);**

Назначение:

Функция записывает ключ программного датчика случайных чисел (ПДСЧ) в файл или на ТМ.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init()
prnd_filename	in	имя файла, куда будет записан ключ ПДСЧ, или NULL, если ключ хранится на ТМ-идентификаторе

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_WRITE_FILE	Ошибка записи файла
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_BAD_CRYP	Ошибка шифрования

Описание функции:

Функция записывает ключ программного датчика случайных чисел в файл или ТМ. Последующий вызов функции cr_init_prnd будет брать данные из этого файла. Файл должен располагаться на отчуждаемом носителе (дискете или флешке).

Примечание:

Данная функция реализована в криптобиблиотеках в вариантах исполнений 1, 2, 15, 16.

1.11. Загрузка дополнительных библиотек

int DECL **cr_load_bicr_dll** (**char** **dll_path)*

Назначение:

Функция загружает библиотеку grn.dll, grn64.dll, libbiogrn.dylib или libbiogrn_64.dylib по указанному пути.

Параметры:

dll_path	in	Путь к каталогу, в котором хранятся библиотеки grn.dll, grn64.dll или libbicr5exp.dylib, libbiogrn.dylib

Возвращаемое значение:

Функция возвращает код ошибки, который должен интерпретироваться как предупреждение:

ERR_OK	Ошибок не обнаружено
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки

Описание функции:

Функция загружает библиотеки grn.dll, grn64.dll, libbiogrn.dylib или libbiogrn_64.dylib из указанного в параметре каталога.

1.12. Универсальная функция инициализации

```
#ifdef SIGN_ONLY
int DECL cr_init_bicr4(char *dll_path,
                       int tm_flag,
                       int *init_mode,
                       H_INIT *init_handle,
                       char *prnd_filename,
                       int flag_init_grn,
                       int  *warning);
#else
int DECL cr_init_bicr4 (char *dll_path,
                       int tm_flag,
                       const char *gk,
                       const char *uz,
                       const char *psw,
                       void *tm_number,
                       int *tmn_blen,
                       int *init_mode,
                       H_INIT *init_handle,
                       char *prnd_filename,
                       int flag_init_grn,
                       int  *warning);
```

Назначение:

Данная функция обеспечивает возможность последовательного вызова сразу трех функций:

- cr_load_bicr_dll
- cr_init
- cr_init_prnd

с одним набором параметров.

Параметры:

dll_path	in	Путь к каталогу, в котором хранятся библиотеки bicr5.dll, grn.dll или libbicr5exp.dylib, libbiogrn.dylib
tm_flag	in	0 – если не требуется использовать специализированные носители ключевой информации 1 - если в дальнейшем предполагается работа с ТМ-идентификатором
gk, uz	in	путь к файлам с главным ключом и узлом замены; если носителем главного ключа и узла замены является ТМ, данные параметры должны принимать значение NULL. Если для работы использование главного ключа и узла замены не требуется, то данный параметр можно установить в "\0" или в "". В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
psw	in	пароль, на котором зашифрован главный ключ при хранении; если главный ключ хранится в открытом виде - данный параметр должен принимать значение "\0". В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
tm_number	out	указатель на буфер, в который возвращается номер ТМ. Если носителем ключевой информации является файл (дискета), данный параметр игнорируется В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе – реальную возвращенную длину буфера tm_number если носителем ключевой информации является файл (дискета), данный параметр игнорируется В версии без шифрования (SIGN_ONLY) этот параметр не требуется.
init_mode	out	возвращаемое значение, которое указывает на наличие драйверов, используемых в процессе работы библиотеки, и результат ввода главного ключа init_mode = cGKUZ | cDSCH | cTMDRV | cVPNKEY Значения соответствующих флажков в битовом представлении таковы: cGKUZ 00000001 главный ключ корректно введен cDSCH 00000010 ДСЧ обнаружен cTMDRV 00000100 драйвер работы с ТМ обнаружен cVPNKEY[4] 00010000 токен обнаружен
init_handle	out	возвращаемый дескриптор H_INIT, используемый другими функциями библиотеки, характеризующий контекст библиотеки. В случае неустранимой ошибки устанавливается в NULL. В случае простой ошибки возвращается ненулевое значение и с библиотекой можно работать.
prnd_filename	in	имя файла, где хранится ключ ПДСЧ, или NULL, если ключ хранится на ТМ-идентификаторе файл должен располагаться на отчуждаемом носителе (дискете или флешке)
flag_init_grn	in	В случае ошибки при чтении файла с ключом ПДСЧ: если flag_init_grn = 1, то для инициализации ПДСЧ будет автоматически использован ПДСЧ из библиотеки grn.dll, grn64.dll, libbiogrn.dylib или libbiogrn_64.dylib если flag_init_grn = 0, то библиотеки grn.dll, grn64.dll, libbiogrn.dylib и libbiogrn_64.dylib не будут использоваться и будет выдана ошибка
warning	out	здесь будет записано значение, которое вернул вызов cr_init при этом init_handle должен быть не равен нулю, то есть инициализация прошла успешно, но есть неполная инициализация, например, ГК не был загружен

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

[4] Не используется при работе в ОС macOS.

ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_OPEN_FILE	Ошибка открытия файла
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_DSCH	Ошибка ПДСЧ
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYP	Ошибка шифрования
ERR_READ_FILE	Ошибка чтения файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция является комбинацией трех функций

- cr_load_bicr_dll
- cr_init
- cr_init_prnd

вызываемых последовательно. Возвращаемое значение функции cr_init записывается в переменную warning. Код ошибки из cr_init_bicr4 возвращается только в случае неустранимой ошибки, например, если tm_flag установить равным 1 (требуется драйвер работы с ТМ-идентификатором TMDRV), но драйвер TMDRV не установлен в системе.

2 Функции вычисления значений хеш-функции

Библиотека обеспечивает вычисление значения хеш-функции в соответствии с алгоритмом ГОСТ Р 34.11-2012 "Информационная технология. Криптографическая защита информации. Функция хэширования". Необходимо использовать функцию cr_set_param() с константами 'F' или 'G' из Приложения 3.
Набор функций, предназначенных для подсчета значения хеш-функции для блока данных, состоит из функций cr_hash_open(), cr_set_param(), cr_hash_calc(), cr_hash_return() и cr_hash_close(). Данный набор функций следует использовать для контроля целостности массивов данных, а также при формировании и проверке ЭП для блоков данных большого размера.

2.1. Инициализация вычислений значений хеш-функции

int DECL **cr_hash_open(H_HASH** **hash_handle* **);**

Назначение:

Функция возвращает дескриптор H_HASH для дальнейшего вычисления значения хеш-функции какого-либо информационного буфера.

Параметры:

hash_handle	out	возвращаемый дескриптор H_HASH, характеризующий промежуточные значения хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Данная функция должна быть вызвана первой из набора функций, предназначенных для подсчета значения хеш-функции для буфера памяти. Функция возвращает дескриптор H_HASH, характеризующий промежуточные значения хеш-функции.
После вызова cr_hash_open() для расчета значения хеш-функции должны быть вызваны cr_hash_calc(), cr_hash_return() и cr_hash_close().

2.2. Вычисление значения хеш-функции для части блока данных

int DECL **cr_hash_calc(H_HASH** *hash_handle*,
void **buf*,
int *buf_len* **);**

Назначение:

Подсчитывает значение хеш-функции для одной из частей блока данных большого размера.

Параметры:

hash_handle	in	дескриптор H_HASH, возвращаемый функцией cr_hash_open(), характеризующий промежуточные значения хеш-функции
buf	in	указатель на буфер, в котором содержится часть общего блока данных
buf_len	in	размер буфера данных

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Данная функция позволяет подсчитать хеш-функцию для блока данных путем подсчета значений хеш-функции для отдельных его частей.

- Пример использования 1. Создать с помощью функции cr_hash_open() дескриптор H_HASH, затем в цикле последовательно один за другим размещать в буфере buf части блока данных и вызывать функцию cr_hash_calc() до тех пор, пока весь блок данных не будет обработан. В параметре buf_len должна передаваться реальная длина обрабатываемой части блока данных. Эту особенность следует учитывать при обработке последней части, если размер блока данных не кратен размеру выделенного буфера buf. После обработки последней части необходимо вызвать функцию cr_hash_return(), которая завершит подсчет значения хеш-функции для всего блока данных и вернет окончательное значение. В конце работы вызвать cr_hash_close для деинициализации.
- Пример использования 2. Создать с помощью функции cr_hash_open() дескриптор H_HASH, вызвать функцию cr_hash_calc()для всего блока данных и вызвать функцию cr_hash_return(), которая завершит подсчет значения хеш-функции и вернет окончательное значение. В конце работы вызвать cr_hash_close для деинициализации.

2.3. Завершение вычисления значения хеш-функции для блока данных

int DECL **cr_hash_return(H_HASH** *hash_handle,*
void **hash_result,*
int ***hash_blen*);**

Назначение:

Функция завершает подсчет значения хеш-функции для всего блока данных и возвращает окончательное значение.

Параметры:

hash_handle	in	дескриптор H_HASH, характеризующий промежуточные значения хеш-функции
hash_result	out	указатель на буфер, в который возвращается результат вычисления значения хеш-функции

hash_blen	in/out	на входе содержит максимальную длину буфера hash_result, на выходе - реальную возвращенную длину буфера hash_result (32 байта)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Данная функция возвращает окончательное значение хеш-функции как результат обработки отдельных частей блока данных. Результат вычисления значения хеш-функции записывается в hash_result. Данную функцию следует вызывать и в том случае, если блок данных обработан за один вызов функции cr_hash_calc().
На основании значения хеш-функции можно сформировать ЭП, используя функцию cr_sign_hash(), или проверить ЭП, используя функцию cr_check_hash().

- Примечание 1. Если параметр hash_result при вызове функции имел значение NULL, то функция не возвращает значения хеш-функции, но оно сохраняется в структуре hash_handle до вызова функции cr_hash_close().
- Примечание 2. На входе необходимо задать длину начального буфера hash_result, реальный размер сформированного буфера возвращается в параметре hash_blen. Если функция возвращает код ошибки ERR_MORE_DATA, то через параметр hash_blen возвращается необходимая длина буфера.

2.4. Установка значения хеш-функции

int DECL **cr_hash_set(** **H_HASH** *hHash,*
const char *hash[32]*);

Назначение:

Функция устанавливает значение хеш-функции для дескриптора hHash.

Параметры:

hHash	in/out	дескриптор H_HASH, характеризующий промежуточные значения хеш-функции
hash	in	указатель на буфер, который содержит значение хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Функция устанавливает значение хеш-функции для дескриптора hHash. Таким образом любые функции, работающие с дескриптором хеш-функции будут принимать заданное значение.

2.5. Создание копии дескриптора хеш-функции

int DECL **cr_hash_dup(** **H_HASH** *hHashSrc,*
H_HASH * *hHashDst* **);**

Назначение:

Функция копирует дескриптор хеш-функции.

Параметры:

hHashSrc	in	копируемый дескриптор H_HASH, характеризующий промежуточные значения хеш-функции
hHashDst	out	копия дескриптора hHashSrc

Возвращаемое значение:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Функция копирует дескриптор хеш-функции. Таким образом можно закрыть исходный дескриптор и получить значение хеш-функции, а при помощи скопированного дескриптора можно продолжить вычисление значения хеш-функции.

2.6. Закрытие дескриптора хеш-функции

int DECL cr_hash_close (**H_HASH** *Ahash_struct);*

Назначение:

Закрытие дескриптора хеш-функции.

Параметры:

Ahash_struct	in	Дескриптор хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный дескриптор библиотеки, например, дескриптор был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Закрывает дескриптор хеш-функции по окончании работы.

3 Функции формирования и загрузки ключей ЭП

3.1. Генерация мастер-ключа

int DECL **cr_gen_masterkey(H_INIT** *init_handle***,**
void **masterkey***,*
int **master_blen***);*

Назначение:

Функция создает мастер-ключ (маску) для хранения ключей ЭП пользователей.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	Проинициализированный ранее дескриптор H_INIT, H_USER, H_PKEY или H_HASH
masterkey	out	указатель на буфер, в который возвращается созданный мастер-ключ;
master_blen	in/out	на входе содержит максимальную длину буфера masterkey, на выходе - реальную возвращенную длину буфера masterkey (108 байт)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования

Описание функции:

Функция создает мастер-ключ, который используется при хранении ключей ЭП пользователей. Мастер-ключ возвращается в открытом виде. Он должен быть записан на внешний носитель и сохранен в надежном месте. Для использования мастер-ключа, его необходимо установить на локальный компьютер (зашифровать на главном ключе) с помощью функции cr_install_masterkey().
Функция доступна, если в результате вызова функции cr_init() установлен флаг cDSCH.
Мастер ключ может быть единым для всех пользователей системы.
Мастер ключ должен храниться непосредственно на компьютере, на котором установлена система, в виде файла или в виде записи в базе данных для обеспечения невозможности использования ключа ЭП пользователя вне системы.

3.2. Установка мастер-ключа

int DECL **cr_install_masterkey(H_INIT** *init_handle*,
void **masterkey*,
int *master_blen*,
void **masterkey_gk*,
int **master_blen_gk*);

Назначение:

Функция устанавливает (зашифровывает на главном ключе) заранее созданный мастер-ключ для хранения ключей ЭП пользователей.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	Проинициализированный ранее дескриптор H_INIT, H_USER, H_PKEY или H_HASH
masterkey	in	указатель на буфер с мастер-ключом, созданный функцией cr_gen_masterkey()
master_blen	in	содержит длину буфера masterkey
masterkey_gk	out	указатель на буфер, в который возвращается мастер-ключ, зашифрованный на главном ключе;

master_blen_gk	in/out	на входе содержит максимальную длину буфера masterkey_gk, на выходе – реальную возвращенную длину буфера masterkey_gk

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_CRYP	Ошибка шифрования
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ

Описание функции:

Функция создает мастер-ключ, который используется при хранении ключей ЭП пользователей.
Мастер-ключ возвращается зашифрованным на главном ключе.
Функция доступна, если в результате вызова функции cr_init() установлены флаги cDSCH и cGKUZ.
Мастер ключ может быть единым для всех пользователей системы.
Мастер ключ должен храниться непосредственно на компьютере, на котором установлена система, в виде файла или в виде записи в базе данных для обеспечения невозможности использования ключа ЭП пользователя вне системы.

3.3. Перешифрование мастер-ключа на новом Главном ключе

int DECL **cr_change_masterkey(H_INIT** *init_handle*,
const char *old_*master_name*,
const char *new_*master_name*,
const char *gk_name,
const char *uz_name,
const char *passw **);**

Назначение:

Функция перешифровывает установленный мастер-ключ на новом Главном ключе. Используется для плановой замены Главного ключа.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	Проинициализированный ранее дескриптор H_INIT, H_USER, H_PKEY или H_HASH
old_master_name	in	имя файла для чтения мастер-ключа, зашифрованного на старом ГК, который был получен при помощи функции cr_install_masterkey()
new_master_name	in	имя нового файла с мастер-ключом, который будет перешифрован на новом ГК и который будет в дальнейшем использоваться в функции cr_load_elgkey_mk()
gk_name	in	имя файла с новым Главным ключом (или NULL, если ключ на ТМ-идентификаторе), старый Главный ключ должен быть загружен в системе
uz_name	in	имя файла с новыми Узлами замены (или NULL, если они находятся на ТМ-идентификаторе)
passw	in	пароль нового Главного ключа

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_BAD_PARAM	не задан параметр init_handle, old_master_name, new_master_name
ERR_INIT	перед вызовом функции не были проинициализированы дескрипторы данных
ERR_OPEN_FILE	файл с мастер-ключом не обнаружен
ERR_WRITE_FILE	ошибка при записи файла
ERR_READ_FILE	ошибка при чтении файла

Описание функции:

Функция перешифровывает на новом Главном ключе установленный ранее при помощи cr_install_masterkey() мастер-ключ. Функция возвращает зашифрованный на новом Главном ключе мастер-ключ и записывает его в файл.

3.4. Генерация ключей ЭП с "привязкой" к ТМ

```
int DECL cr_gen_elgkey( H_INIT init_handle,
                        char *password,
                        int *pass_blen,
                        char *secrkey,
                        int *secr_blen,
                        H_PKEY *pkey_handle,
                        const char *userid,
                        void *tm_number,
                        int *tmn_blen);
```

Назначение:

Функция создает пару (закрытый и открытый) ключей ЭП. Закрытый ключ будет записываться **только** в ТМ-идентификатор.

Параметры:

init_handle	in	Проинициализированный ранее дескриптор H_INIT, H_USER, H_PKEY или H_HASH
password	out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
pass_blen	in/out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
secrkey	out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
secr_blen	in/out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
userid	in	указатель на буфер с идентификатором пользователя, заканчивающийся нулем, длиной не более 32-х байт
tm_number	out	указатель на буфер, в который возвращается номер ТМ если номер не нужен, можно установить данный параметр равным NULL
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number указатель на буфер, в который возвращается номер ТМ если номер не нужен, можно установить данный параметр равным NULL

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки

ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП (закрытый и открытый).
Функция доступна, если в результате вызова функции cr_init() установлен флаг cDSCH.
Ключевая информация может быть записана в ТМ-идентификатор, если в результате вызова функции cr_init() флаг cTMDRV был установлен.
Если в качестве носителя ключевой информации выбрано устройство ТМ (параметр secrkey имеет значение NULL), функция считывает номер ТМ, создает с учетом номера ТМ ключ и производит попытку записать ключевую информацию в ТМ. Если в момент вызова функции ТМ устройство не готово (не приложено к считывателю), возвращается соответствующий код ошибки и функцию генерации ключей следует вызвать повторно.
Если в качестве носителя ключевой информации выбрано устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть загружен с помощью вызова функции cr_load_elgkey().

3.5. Генерация ключей ЭП в файл с паролем

int DECL **cr_gen_keypair(H_INIT** *init_handle***,**
char *****password*,**
int *****pass_blen*,**
char *****ConfidentFile*,**
H_PKEY *****pkey_handle*,**
const char *****userid***);**

Назначение:

Функция создает пару (закрытый и открытый) ключей ЭП.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
password	out	указатель на буфер, в который возвращается пароль доступа к закрытому ключу ЭП;
pass_blen	in/out	на входе содержит максимальную длину буфера password, на выходе - реальную возвращенную длину буфера password
ConfidentFilename	out	имя файла с носителем ключевой информации, файл должен располагаться на отчуждаемом носителе (USB-флеш диск или дискета);
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
userid	in	указатель на буфер с идентификатором пользователя, заканчивающийся нулем, длиной не более 32-х байт

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора

ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП (закрытый и открытый) и записывает ключевую информацию в файл на отчуждаемом носителе.
Функция доступна, если в результате вызова функции cr_init() флаг cDSCH был установлен.
В качестве меры защиты от копирования и несанкционированного использования ключевой информации создается пароль доступа к этой информации.
Пароль случайно выбирается из множества символов 0-9, a-z, A-Z и имеет длину 6 символов, из которых 5 символов - собственно пароль и один символ - контрольное значение (CRC) для верификации ввода пароля. При этом пароль не может содержать символы: '0', 'O', '1', 'l', 'I', 'B', '8', 'G', которые легко перепутать между собой из-за сходного начертания.
Данный пароль должен быть отображен на экране в момент генерации ключей.
Если владелец ключей считает пароль слишком сложным для запоминания, то ему должна быть предоставлена возможность генерации новой пары ключей до тех пор, пока он не согласится с предложенным паролем.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть загружен с помощью вызова функции cr_read_skey().

3.6. Генерация ключей ЭП с маской - мастер-ключом

int DECL **cr_gen_elgkey_mk(H_INIT** *init_handle***,**
const void ***_masterkey_,**
int *master_len***,**
void ***_secrkey_,**
**int ***_secr_blen,_
H_PKEY ***_pkey_handle_,**
const char ***_userid_,**
void ***_tm_number_,**
int ***_tmn_blen_);**

Назначение:

Функция производит генерацию пары (закрытый и открытый) ключей ЭП пользователя. При хранении закрытого ключа ЭП используется мастер-ключ (см. cr_ install_masterkey()). Использовать **только** при работе с ТМ-идентификатором.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
masterkey	in	указатель на буфер, в котором содержится "установленный" мастер-ключ

master_len	in	размер буфера masterkey
secrkey	out	при генерации ключа на ТМ установите данный параметр в NULL
secr_blen	in/out	при генерации ключа на ТМ установите данный параметр в NULL
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
userid	in	указатель на буфер с идентификатором пользователя, заканчивающийся нулем, длиной не более 32-х байт
tm_number	out	указатель на буфер, в который возвращается номер ТМ. если номер не нужен, можно установить данный параметр равным NULL
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе – реальную возвращенную длину буфера tm_number если номер не нужен, можно установить данный параметр равным NULL

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки

ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП пользователя (закрытый и открытый) и записывает ключевую информацию непосредственно в ТМ-идентификатор (параметр secrkey имеет значение NULL).
Функция доступна, если в результате вызова функции cr_init() установлены флаги cDSCH и cGKUZ.
Ключевая информация может быть записана в ТМ-идентификатор, если в результате вызова функции cr_init() флаг cTMDRV установлен.
При хранении закрытого ключа ЭП используется мастер-ключ, который должен быть до вызова данной функции создан с помощью функции cr_gen_masterkey() и установлен (зашифрован на главном ключе) с помощью функции cr_install_masterkey().
Допускается использование одного мастер-ключа для всех пользователей системы.
Если в качестве носителя ключевой информации выбрано устройство ТМ, функция создает ключ и производит попытку записать ключевую информацию в ТМ. Если в этот момент ТМ устройство не готово (не приложено к считывателю), возвращается соответствующий код ошибки и функцию генерации ключей следует вызвать повторно.
Если в качестве носителя ключевой информации выбрано устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть явно загружен в него с помощью вызова функции cr_load_elgkey_mk().

3.7. Генерация ключей ЭП в файл с маской - мастер-ключом

```
int DECL cr_gen_keypair_mk( H_INIT init_handle,
                            const void *masterkey,
                            int master_len,
                            char *ConfidentFilename,
                            H_PKEY *pkey_handle,
                            const char *userid);
```

Назначение:

Функция производит генерацию пары (закрытый и открытый) ключей ЭП пользователя. При хранении закрытого ключа ЭП используется мастер-ключ (см. cr_ install_masterkey()).

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
masterkey	in	указатель на буфер, в котором содержится "установленный" мастер-ключ
master_len	in	размер буфера masterkey
ConfidentFilename	out	имя файла с носителем ключевой информации, файл должен располагаться на отчуждаемом носителе (USB-флеш диск или дискета);
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
userid	in	указатель на буфер с идентификатором пользователя, заканчивающийся нулем, длиной не более 32-х байт

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения TM-идентификатора
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_WRITE_TM	Ошибка записи TM-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_NO_TM_ATTACHED	TM-идентификатор не приложен к съемнику

ERR_CRC_TM	Ошибка контрольной суммы TM-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП пользователя (закрытый и открытый) и записывает ключевую информацию в файл на отчуждаемом носителе.
Функция доступна, если в результате вызова функции cr_init() установлены флаги cDSCH и cGKUZ.
При хранении закрытого ключа ЭП используется мастер-ключ, который должен быть до вызова данной функции создан с помощью функции cr_gen_masterkey(), а затем установлен (зашифрован на главном ключе) с помощью функции cr_install_masterkey().
Допускается использование одного мастер-ключа для всех пользователей системы.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть явно загружен в него с помощью вызова функции cr_read_skey_mk().

3.8. Загрузка закрытого ключа ЭП с паролем или "привязкой" к ТМ

int DECL **cr_load_elgkey(H_INIT** *init_handle*,
char **password*,
int *passblen*,
char **secrkey*,
int *secr_blen*,
**void ***tm_number,*
**int ***tmn_blen,*
char **userid,*
int **userid_blen,*
**H_USER ***user_handle*);

Назначение:

Функция загружает закрытый ключ ЭП Администратора, созданный с помощью функции cr_gen_elgkey(), в контекст библиотеки. Использовать **только** при работе с ТМ-идентификатором.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
password	in	при чтении ключа с ТМ или VPN-Key установите данный параметр в NULL
passblen	in	при чтении ключа с ТМ или VPN-Key установите данный параметр равным 0
secrkey	in	при чтении ключа с ТМ или VPN-Key установите данный параметр в NULL
secr_blen	in	при чтении ключа с ТМ или VPN-Key установите данный параметр равным 0
tm_number	out	указатель на буфер, в который возвращается номер ТМ если номер не нужен, можно установить данный параметр равным NULL

tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number если номер не нужен, можно установить данный параметр равным NULL
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца закрытого ключа ЭП
userid_blen	in/out	на входе содержит максимальную длину идентификатора пользователя, на выходе реальную возвращенную длину
user_handle	out	возвращаемый дескриптор H_USER, характеризующий закрытый ключ ЭП пользователя

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_OPEN_FILE	Ошибка открытия файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция считывает с ключевого носителя (дискеты или ТМ) ключевую информацию, созданную с помощью функции cr_gen_elgkey(), формирует закрытый ключ (с учетом пароля или номера ТМ) и загружает закрытый ключ ЭП в контекст библиотеки. Ссылка на закрытый ключ в контексте библиотеки возвращается через параметр user_handle, что позволяет загружать в контекст несколько закрытых ключей. В случае ошибки в процессе загрузки ключа, параметр user_handle устанавливается в NULL.
Ключевая информация может быть считана из ТМ-идентификатора, если в результате вызова функции cr_init() флаг cTMDRV установлен.
Если носителем ключевой информации является устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
Если в момент вызова функции устройство не готово (ТМ-идентификатор не присоединен к считывателю), возвращается соответствующий код ошибки и функцию следует вызвать повторно.

3.9. Загрузка из файла закрытого ключа ЭП с паролем

int DECL **cr_read_skey(H_INIT** *init_handle*,
char **password*,
int *passblen*,
char **ConfidentFilename*,
char **userid,*
int **userid_blen,*
H_USER * *user_handle*);

Назначение:

Функция загружает закрытый ключ ЭП, созданный с помощью функции cr_gen_keypair(), в контекст библиотеки.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
password	in	указатель на буфер, в котором содержится пароль, защищающий доступ к закрытому ключу ЭП;
passblen	in	длина строки с паролем
ConfidentFilename	in	имя файла с носителем ключевой информации, файл должен располагаться на отчуждаемом носителе (USB-флеш диск или дискета);
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца закрытого ключа ЭП
userid_blen	in/out	на входе содержит максимальную длину идентификатора пользователя, на выходе реальную возвращенную длину
user_handle	out	возвращаемый дескриптор H_USER, характеризующий закрытый ключ ЭП пользователя

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее

ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_OPEN_FILE	Ошибка открытия файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция считывает с ключевого носителя (USB-флеш диска или дискеты) ключевую информацию, созданную с помощью функции cr_gen_keypair(), формирует закрытый ключ (с учетом пароля) и загружает закрытый ключ ЭП в контекст библиотеки. Ссылка на закрытый ключ в контексте библиотеки возвращается через параметр user_handle, что позволяет загружать в контекст несколько закрытых ключей. В случае ошибки в процессе загрузки ключа, параметр user_handle устанавливается в 0.
Функция доступна всегда.

3.10. Загрузка закрытого ключа ЭП с мастер-ключом

```
int DECL cr_load_elgkey_mk(H_INIT init_handle,
                           void *masterkey,
                           int master_blen,
                           char *secrkey,
                           int secr_blen,
                           void *tm_number,
                           int *tmn_blen,
                           char *userid,
                           int *userid_blen,
                           H_USER *user_handle);
```

Назначение:

Функция загружает закрытый ключ ЭП пользователя, созданный с помощью функции cr_gen_elgkey_mk(), в контекст библиотеки. Использовать **только** при работе с ТМ-идентификатором.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
master_key	in	указатель на буфер, в котором содержится "установленный" мастер-ключ;
master_blen	in	размер буфера master_key
secrkey	in	при чтении ключа с ТМ или VPN-Key установите данный параметр в NULL
secr_blen	in	при чтении ключа с ТМ или VPN-Key установите данный параметр равным 0
tm_number	out	указатель на буфер, в который возвращается номер ТМ если номер не нужен, можно установить данный параметр равным NULL
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number если номер не нужен, можно установить данный параметр равным NULL
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца закрытого ключа ЭП
userid_blen	in/out	на входе содержит максимальную длину идентификатора пользователя, на выходе реальную возвращенную длину
user_handle	out	возвращаемый дескриптор H_USER, характеризующий закрытый ключ пользователя

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна

ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_OPEN_FILE	Ошибка открытия файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция загружает закрытый ключ ЭП, созданный с помощью функции cr_gen_elgkey_mk(), в контекст библиотеки. Буфер master_key должен содержать мастер-ключ "установленный" (зашифрованный на главном ключе) с помощью функции cr_install_masterkey(). Ссылка на закрытый ключ в контексте библиотеки возвращается через параметр user_handle, что позволяет загружать в контекст несколько закрытых ключей. В случае ошибки в процессе загрузки ключа, параметр user_handle устанавливается в NULL.
Функция доступна всегда, однако, если в результате вызова функции cr_init() не установлены флаги cDSCH или cGKUZ, считывание ключевой информации с носителя не производится, но возвращается номер ТМ (если параметр secrkey указывает на ТМ, как на носитель ключевой информации).
Ключевая информация может быть считана из ТМ-идентификатора, если в результате вызова функции cr_init() флаг cTMDRV установлен.
Если носителем ключевой информации является устройство ТМ, то функция возвращает номер ТМ. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
Если в момент вызова функции устройство не готово (ТМ-идентификатор не присоединен к считывателю), возвращается соответствующий код ошибки и функцию следует вызвать повторно.

3.11. Загрузка из файла закрытого ключа ЭП с мастер-ключом

```
int DECL cr_read_skey_mk(H_INIT init_handle,
                         void *masterkey,
                         int master_blen,
                         char *ConfidentFilename,
                         char *userid,
                         int *userid_blen,
                         H_USER *user_handle);
```

Назначение:

Функция загружает закрытый ключ ЭП пользователя, созданный с помощью функции cr_gen_keypair_mk(), в контекст библиотеки.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
master_key	in	указатель на буфер, в котором содержится "установленный" мастер-ключ;

master_blen	in	размер буфера master_key
ConfidentFilename	in	имя файла с носителем ключевой информации, файл должен располагаться на отчуждаемом носителе (USB-флеш диск или дискета);
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца закрытого ключа ЭП
userid_blen	in/out	на входе содержит максимальную длину идентификатора пользователя, на выходе реальную возвращенную длину
user_handle	out	возвращаемый дескриптор H_USER, характеризующий закрытый ключ пользователя

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_OPEN_FILE	Ошибка открытия файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция загружает закрытый ключ ЭП, созданный с помощью функции cr_gen_keypair_mk(), в контекст библиотеки. Буфер master_key должен содержать мастер-ключ "установленный" (зашифрованный на главном ключе) с помощью функции cr_install_masterkey(). Ссылка на закрытый ключ в контексте библиотеки возвращается через параметр user_handle, что позволяет загружать в контекст несколько закрытых ключей. В случае ошибки в процессе загрузки ключа, параметр user_handle устанавливается в NULL.
Функция доступна всегда.

3.12. Получение идентификатора закрытого ключа ЭП

int DECL **cr_elgkey_getid**(**H_INIT** *init_handle*,
H_USER *user_handle,*
**char ***userid,*
**int ***userid_blen*);

Назначение:

Функция возвращает идентификатор закрытого ключа ЭП, загруженного в контекст библиотеки с помощью функции cr_load_elgkey() или cr_load_elgkey_mk().

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey() или cr_load_elgkey_mk() и характеризующий закрытый ключ пользователя в контексте библиотеки
userid	out	указатель на строку, через которую возвращается идентификатор закрытого ключа (идентификатор возвращается с заключительным нулем)
userid_blen	in/out	на входе - максимально доступный размер строки; на выходе - реальный размер возвращенной строки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция возвращает идентификатор закрытого ключа пользователя, характеризуемого параметром user_handle.

3.13. Выгрузка закрытого ключа ЭП

int DECL **cr_elgkey_close**(H_INIT *init_handle*,
H_USER *user_handle*);

Назначение:

Функция выгружает (освобождает ранее отведенную память и затирает информацию в памяти) закрытый ключ ЭП, загруженный с помощью функции cr_load_elgkey(), cr_load_elgkey_mk(), cr_read_skey() или cr_read_skey_mk() из контекста библиотеки. Если далее пытаться пользоваться выгруженным ключом, в функциях подписи происходит ошибка ERR_BAD_HANDLE.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey() или cr_load_elgkey_mk() и характеризующий закрытый ключ пользователя в контексте библиотеки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Функция выгружает из контекста библиотеки закрытый ключ пользователя, характеризуемый дескриптором user_handle.

3.14. Генерация открытого ключа

int DECL **cr_gen_pubkey(H_INIT** *init_handle***,**
H_USER *user_handle***,**
H_PKEY **pkey_handle***);*

Назначение:

Функция создает открытый ключ на основе загруженного закрытого ключа.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, с загруженным закрытым ключом.
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция создает открытый ключ, соответствующий закрытому ключу и возвращает его дескриптор.

3.15. Генерация главного ключа

int DECL **cr_gen_gk(H_INIT** *init_handle*,
char **password*,
void **gkbuf*,
int **gkblen*
**void ***tm_number,*
**int ***tmn_blen*);

Назначение:

Функция создает новый Главный ключ (GK).

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
password	in	указатель на буфер, в котором содержится пароль доступа к Главному ключу; если пользователь не задал пароль, данный параметр принимает значение "\0" ("пустая" строка)
gkbuf	out	указатель на буфер, в который возвращается зашифрованный на пароле Главный ключ данный параметр определяет носитель ключевой информации: • если данный параметр указывает на какой-либо буфер (имеет ненулевое значение), то носителем ключевой информации является файл (дискета); • если данный параметр имеет значение NULL, то носителем ключевой информации является ТМ-идентификатор;
gkblen	in/out	если носителем ключевой информации является файл (дискета), параметр на входе содержит максимальную длину буфера gkbuf, на выходе - реальную возвращенную длину буфера gkbuf если носителем ключевой информации является ТМ-идентификатор, параметр игнорируется

tm_number	out	указатель на буфер, в который возвращается номер ТМ. если носителем ключевой информации является файл (дискета), данный параметр игнорируется
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number если носителем ключевой информации является файл (дискета), данный параметр игнорируется

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику

Описание функции:

Функция создает новый Главный ключ, шифрует его на пароле, введенном пользователем, и в зависимости от значения параметра gkbuf записывает его либо в файл (для последующего хранения на ключевой дискете), либо непосредственно в ТМ-идентификатор.
Функция доступна, если в результате вызова функции cr_init() флаг cDSCH установлен.
Ключевая информация может быть записана в ТМ-идентификатор, если в результате вызова функции cr_init() флаг cTMDRV установлен.
Пароль, на котором шифруется Главный ключ, должен быть введен пользователем до вызова функции.
Если в качестве носителя ключевой информации выбрано устройство ТМ (параметр gkbuf имеет значение NULL), функция считывает номер ТМ и производит попытку записать ключевую информацию в ТМ. Если в момент вызова функции ТМ устройство не готово (не приложено к считывателю), возвращается соответствующий код ошибки и функцию генерации ключей следует вызвать повторно.
Если в качестве носителя ключевой информации выбрано устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.

После генерации новый Главный ключ не сохраняется в контексте библиотеки, для его использования библиотека должна быть деинициализирована с помощью функции cr_uninit() и повторно инициализирована с помощью вызова функции cr_init(), при этом должен быть введен новый Главный ключ.

3.16. Копирование главного ключа в ТМ

int DECL **cr_gk_copy_tm(H_INIT** *init_handle***,**
void **gkbuf***,*
int **gkblen*
char ***tm_number,*
int ***tmn_blen***);**

Назначение:

Функция копирует Главный ключ из буфера в ТМ-идентификатор.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
gkbuf	in	указатель на буфер, в котором содержится (возможно, зашифрованный на пароле) главный ключ.
gkblen	in/out	на входе содержит максимальную длину буфера gkbuf, на выходе - реальную возвращенную длину буфера gkbuf
tm_number	out	указатель на буфер, в который возвращается номер ТМ
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику

Описание функции:

Копирует ГК из буфера в ТМ-идентификатор. В ТМ-идентификатор также записываются Узлы замены. Узлы замены берутся те, которые были загружены в память при вызове cr_init(). Если ГК был защищен паролем, то ГК на ТМ-идентификаторе также будет защищен тем же паролем.

3.17. Расширенная генерация ключей ЭП с паролем или “привязкой” к ТМ

int DECL **cr_gen_elgkey_ext**(**H_INIT** *init_handle*,
const char **userid*,
char **password*,
int **pass_blen,*
void **secrkey*,
int **secr_blen,*
int *mode,*
H_PKEY **pkey_handle*,
void **tm_number,*
int **tmn_blen*);

Назначение:

Функция создает основную или резервную пару (закрытый и открытый) ключей ЭП. Использовать **только** при работе с ТМ-идентификатором или VPN-Key. VPN-Key не используется при работе в ОС macOS.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
userid	in	идентификатор пользователя, длиной не более 32-х байт
password	out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
pass_blen	in/out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
secrkey	out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
secr_blen	in/out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
mode	in	параметр определяет, какая пара ключей (основная или резервная) создается • если параметр secrkey принимает нулевое значение (носителем ключевой информации является ТМ-идентификатор) и параметр mode установлен в 0, функция создает закрытый ключ и записывает его в ТМ-идентификатор на место основного ключа ЭП; • если параметр secrkey принимает нулевое значение (носителем ключевой информации является ТМ-идентификатор) и параметр mode установлен в 1, функция создает закрытый ключ и записывает его в ТМ-идентификатор на место резервного ключа ЭП;
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
tm_number	out	указатель на буфер, в который возвращается номер ТМ. если номер не нужен, можно установить данный параметр равным NULL
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number если номер не нужен, можно установить данный параметр равным NULL

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП (закрытый и открытый) и записывает ключевую информацию непосредственно в ТМ-идентификатор, либо в устройство VPN-Key. В ТМ-идентификатор или VPN-Key ключевая информация в зависимости от значения параметра mode записывается на место основного или резервного ключа ЭП.
Функция доступна, если в результате вызова функции cr_init() флаг cDSCH установлен.
Ключевая информация может быть записана в ТМ-идентификатор, если в результате вызова функции cr_init() флаг cTMDRV установлен.
Если в качестве носителя ключевой информации выбрано устройство ТМ или VPN-Key (параметр secrkey имеет значение NULL), функция считывает номер ТМ или VPN-Key, создает с учетом номера ключ и производит попытку записать ключевую информацию в ТМ или VPN-Key. Если в момент вызова функции ТМ устройство не готово (ТМ-идентификатор не присоединен к считывателю или USB-токен не установлен в USB-порт), возвращается соответствующий код ошибки и функцию генерации ключей следует вызвать повторно. Если параметр mode установлен в 0, ключевая информация записывается в устройство ТМ или VPN-Key на место основного ключа ЭП. Если параметр mode установлен в 1, ключевая информация записывается в устройство на место резервного ключа ЭП.
Если параметр secrkey принимает нулевое значение и параметр mode установлен в 0, действие функции cr_gen_elgkey_ext() аналогично действию функции cr_gen_elgkey().
Если в качестве носителя ключевой информации выбрано устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть явно в него загружен. Ключевая информация с дискеты, а также основной ключ ЭП с устройства ТМ должны быть явно загружены в контекст библиотеки с помощью вызова функции "Загрузка закрытого ключа ЭП пользователя". Резервный ключ ЭП с устройства ТМ должен быть предварительно перемещен на место основного ключа с помощью вызова функции "Заменить закрытый ключ ЭП", а затем загружен в контекст библиотеки с помощью функции" Загрузка закрытого ключа ЭП пользователя.

3.18. Расширенная генерация ключей ЭП с маской - мастер-ключом

```
int DECL cr_gen_elgkey_mk_ext(H_INIT init_handle,
                              const char *userid,
                              const void *masterkey,
                              int master_len,
                              void *secrkey,
                              int *secr_blen,
                              int mode,
                              H_PKEY *pkey_handle,
                              void *tm_number,
                              int *tmn_blen);
```

Назначение:

Функция производит генерацию основной или резервной пары (закрытый и открытый) ключей ЭП пользователя. При хранении закрытого ключа ЭП используется мастер-ключ. Использовать **только** при работе с ТМ-идентификатором или VPN-Key.

- Примечание. Данной функции нет в версии библиотеки без шифрования (SIGN_ONLY)

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
userid	in	идентификатор пользователя, длиной не более 32-х байт
masterkey	in	указатель на буфер, в котором содержится мастер-ключ
master_len	in	размер буфера masterkey
secrkey	out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL

secr_blen	in/out	при генерации ключа на ТМ или VPN-Key установите данный параметр в NULL
mode	in	параметр определяет, какая пара ключей (основная или резервная) создается • если параметр secrkey принимает нулевое значение (носителем ключевой информации является ТМ-идентификатор) и параметр mode установлен в 0, функция создает закрытый ключ и записывает его в ТМ-идентификатор на место основного ключа ЭП; • если параметр secrkey принимает нулевое значение (носителем ключевой информации является ТМ-идентификатор) и параметр mode установлен в 1, функция создает закрытый ключ и записывает его в ТМ-идентификатор на место резервного ключа ЭП;
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ ЭП
tm_number	out	указатель на буфер, в который возвращается номер ТМ. если номер не нужен, можно установить данный параметр равным NULL
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе – реальную возвращенную длину буфера tm_number если номер не нужен, можно установить данный параметр равным NULL

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает пару ключей ЭП пользователя (закрытый и открытый) и записывает ключевую информацию непосредственно в ТМ-идентификатор. В ТМ-идентификатор ключевая информация в зависимости от значения параметра mode записывается на место основного или резервного ключа ЭП.
Функция доступна, если в результате вызова функции cr_init() установлены флаги cDSCH и cGKUZ.
Ключевая информация может быть записана в ТМ-идентификатор, если в результате вызова функции cr_init() флаг cTMDRV установлен.
При хранении закрытого ключа ЭП используется мастер-ключ, который должен быть создан до вызова данной функции.
Допускается использование одного мастер-ключа для всех пользователей системы.
Если в качестве носителя ключевой информации выбрано устройство ТМ (параметр secrkey имеет значение NULL), функция создает ключ и производит попытку записать ключевую информацию в ТМ. Если в этот момент ТМ устройство не готово (не приложено к считывателю), возвращается соответствующий код ошибки и функцию генерации ключей следует вызвать повторно. Если параметр mode установлен в 0, ключевая информация записывается в устройство ТМ на место основного ключа ЭП. Если параметр mode установлен в 1, ключевая информация записывается в устройство ТМ на место резервного ключа ЭП.
Если параметр secrkey принимает нулевое значение и параметр mode установлен в 0, действие функции cr_gen_elgkey_mk_ext() аналогично действию функции cr_gen_elgkey_mk().
Если в качестве носителя ключевой информации выбрано устройство ТМ, то функция возвращает номер устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.
После генерации ключей закрытый ключ не сохраняется в контексте библиотеки и должен быть явно в него загружен. Ключевая информация с дискеты, а также основной ключ ЭП с устройства ТМ должны быть явно загружены в контекст библиотеки с помощью вызова функции "Загрузка закрытого ключа ЭП пользователя". Резервный ключ ЭП с устройства ТМ должен быть предварительно перемещен на место основного ключа с помощью вызова функции "Заменить закрытый ключ ЭП", а затем загружен в контекст библиотеки с помощью функции" Загрузка закрытого ключа ЭП пользователя ".

3.19. Замена закрытого ключа ЭП с паролем или “привязкой” к ТМ

int DECL **cr_change_elgkey**(**H_INIT** *init_handle*,
void **tm_number,*
int **tmn_blen,*

char **userid,*
int **userid_blen*);

Назначение:

Функция вводит в действие хранящийся на устройстве ТМ резервный закрытый ключ ЭП Администратора, созданный с помощью функции cr_gen_elgkey_ext(), или резервный закрытый ключ ЭП Пользователя, созданный с помощью функции cr_gen_elgkey_mk_ext().

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
tm_number	out	указатель на буфер, в который возвращается номер ТМ
tmn_blen	in/out	на входе содержит максимальную длину буфера tm_number, на выходе - реальную возвращенную длину буфера tm_number
userid	out	указатель на буфер, в который возвращается идентификатор пользователя – владельца резервного закрытого ключа ЭП
userid_blen	in/out	на входе содержит максимальную длину идентификатора пользователя, на выходе реальную возвращенную длину

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику

Описание функции:

Функция считывает с устройства ТМ резервную ключевую информацию, созданную с помощью функции cr_gen_elgkey_ext() или функции cr_gen_elgkey_mk_ext(), и записывает ее в это же устройство ТМ на место, предназначенное для хранения основного закрытого ключа.
Функция доступна, если в результате вызова функции cr_init() флаг cTMDRV установлен.
Функция возвращает идентификатор владельца резервного ключа и номер ТМ устройства. Номер ТМ устройства может быть использован для идентификации ключевого носителя.

3.20. Создание временного закрытого и открытого ключа ЭП

int DECL **cr_gen_onetime_keypair (H_INIT** *init_handle*,
H_USER ***_user_handle_,**
H_PKEY ***_pkey_handle_,**
char ***userid);**

Назначение:

Функция создает временный закрытый и открытый ключи ЭП пользователя и загружает их в контекст библиотеки. Открытый ключ может быть экспортирован при помощи функции cr_pkey_getinfo(), секретный ключ не экспортируется и должен быть удален в конце сеанса работы.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	out	возвращаемый дескриптор H_USER, характеризующий закрытый ключ пользователя
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ пользователя
userid	in	идентификатор ключа пользователя (не более 32х символов)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция создает временный закрытый и открытый ключи ЭП пользователя и загружает их в контекст библиотеки. После окончания использования следует закрыть соответствующие дескрипторы с помощью cr_elgkey_close и cr_pkey_close.

4 Функции работы со справочником открытых ключей ЭП

Комментарий к открытому ключу - это некоторая структура, которая может быть определена пользователем для каждого ключа и которая сохраняется в базе открытых ключей наряду с бинарной записью открытого ключа и его идентификатором. Размер комментария задается при создании базы открытых ключей и не может быть изменен в последующем.

4.1. Открытие справочника открытых ключей

int DECL **cr_pkbase_open(const char ***_pk_file_**,**
int *com_blen***,**
int *flag_modify***,**
H_PKBASE ***_pkbase_handle_);**

Назначение:

Функция открывает файл, в котором содержится справочник открытых ключей и возвращает дескриптор H_PKBASE, характеризующий данный справочник открытых ключей.

Параметры:

pk_file	in	имя файла справочника
com_blen	in	размер поля, в котором хранится комментарий к открытому ключу
flag_modify	in	равен 1, если база ключей будет в последующем модифицироваться
pkbase_handle	out	возвращаемый дескриптор H_PKBASE, характеризующий справочник открытых ключей

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_PUB	Файла БОК не может быть открыт
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MEM	Недостаточно памяти
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция запускается перед началом работы с данным справочником открытых ключей. Функция доступна всегда. Через параметр pkbase_handle возвращается дескриптор H_PKBASE, характеризующий справочник открытых ключей, который далее используется в функциях работы со справочником.
В одном приложении с помощью данной функции может быть одновременно открыто несколько разных справочников открытых ключей. Также несколько различных приложений одновременно могут использовать несколько справочников открытых ключей.
Комментарии это некоторая структура, которая может быть определена пользователем для каждого ключа и которая сохраняется в базе открытых ключей наряду с бинарной записью открытого ключа и его идентификатором. Размер комментария задается при создании базы открытых ключей и не может быть изменен в последующем.
Если база открытых ключей будет в последующем модифицирована посредством функций cr_pkbase_add() cr_pkbase_remove() и др. тогда следует задать параметр flag_modify равным 1. В этом случае исходная база будет откопирована во временную директорию, и после того как в ней будут сделаны изменения, база с помощью функции cr_pkbase_save() будет откопирована из временной директории и сохранена под именем pk_file.

4.2. Расширенное открытие справочника открытых ключей

int DECL **cr_pkbase_load(H_INIT** *init_handle*,
const char * *pk_file*,
int *com_blen*,
int *flag_modify*,
H_PKBASE * *pkbase_handle*);

Назначение:

Функция открывает файл, в котором содержится справочник открытых ключей и возвращает соответствующий дескриптор, характеризующий данный справочник открытых ключей. Является расширенной версией функции cr_pkbase_open().

Параметры:

pk_file	in	имя файла справочника
com_blen	in	размер поля, в котором записан комментарий к открытому ключу
flag_modify	in	равен 1, если база ключей будет в последующем модифицироваться
pkbase_handle	out	возвращаемый дескриптор H_PKBASE, характеризующий справочник открытых ключей

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_PUB	Файла БОК не может быть открыт
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MEM	Недостаточно памяти
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция запускается перед началом работы с данным справочником открытых ключей. Функция доступна всегда.
Через параметр pkbase_handle возвращается дескриптор H_PKBASE, характеризующий справочник открытых ключей, который далее используется в функциях работы со справочником.
В одном приложении с помощью данной функции может быть одновременно открыто несколько разных справочников открытых ключей. Также несколько различных приложений одновременно могут использовать несколько справочников открытых ключей.
Комментарии - это некоторая структура, которая может быть определена пользователем для каждого ключа и которая сохраняется в базе открытых ключей наряду с бинарной записью открытого ключа и его идентификатором. Размер комментария задается при создании базы открытых ключей и не может быть изменен в последующем.
Если база открытых ключей будет в последующем модифицирована посредством функций cr_pkbase_add() cr_pkbase_remove() и др., тогда следует задать параметр flag_modify равным 1. В этом случае исходная база будет откопирована во временную директорию, и после того как в ней будут сделаны изменения, база с помощью функции cr_pkbase_save() будет откопирована из временной директории и сохранена под тем же именем pk_file.

4.3. Открытие справочника из памяти

int DECL **cr_pkbase_open_membuf(void **membuf*,**
int *mem_blen*,
int *com_blen*,
H_PKBASE **pkbase_handle*);

Назначение:

Параметры:

membuf	in	указатель на буфер памяти, в котором содержится справочник
mem_blen	in	размер буфера памяти membuf
com_blen	in	размер комментария для открытого ключа
pkbase_handle	out	возвращаемый дескриптор H_PKBASE, характеризующий справочник открытых ключей

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти

Описание функции:

Загружает справочник открытых ключей из памяти, в остальном функция аналогична cr_open_pkbase(), как если бы она была бы вызвана с параметром flag_modify = 0, таким образом, буфер модифицировать нельзя.

4.4. Проверка наличия открытого ключа ЭП в справочнике

int DECL **cr_pkbase_find(H_PKBASE** *pkbase_handle,*
const char **userid,*
void ***comment*,**
int ***com_blen*,**
H_PKEY **pkey_handle,***);*

Назначение:

Функция проверяет наличие открытого ключа в справочнике открытых ключей.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, характеризующий справочник открытых ключей
userid	in	идентификатор пользователя владельца открытого ключа
comment	out	указатель на буфер, в который возвращается комментарий к открытому ключу
com_blen	in/out	на входе максимальный размер комментария, на выходе – реальный возвращенный размер комментария comment
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ с данным идентификатором, если он будет найден в справочнике; если задать этот параметр равным NULL, ключ из справочника возвращен не будет, будет возвращен только код ошибки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти

ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция проверяет наличие в справочнике открытых ключей открытого ключа с идентификатором userid и, если он найден, создает и возвращает его дескриптор, который должен быть освобожден функцией cr_pkey_close(). Если параметр pkey_handle задать равным NULL, дескриптор не будет инициализирован и возвращен, даже если ключ найден в справочнике. В этом случае осуществляется только проверка наличия открытого ключа в справочнике и возврат кода ошибки.

4.5. Добавление/замена открытого ключа ЭП в справочник(е)

int DECL **cr_pkbase_add**(**H_PKBASE** *pkbase_handle,*
H_PKEY *cr_pkey_handle*
void ***comment,**
int *com_blen* **);**

Назначение:

Функция добавляет в справочник открытых ключей открытый ключ пользователя. Если ключ пользователя уже существует в справочнике, то он заменяется.

Параметры:

cr_pkbase	in	дескриптор H_PKBASE, характеризующий справочник открытых ключей
cr_pkey_handle	in	дескриптор H_PKEY, характеризующий открытый ключ пользователя
comment	in	комментарий к открытому ключу
com_blen	in	размер комментария для открытого ключа

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_WRITE_FILE	Ошибка записи файла
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция добавляет в справочник, характеризуемый дескриптором H_PKBASE, открытый ключ пользователя, определяемый дескриптором H_PKEY. Если ключ пользователя уже имеется в справочнике, то он заменяется. Чтобы определить, имеется ли открытый ключ пользователя в справочнике, можно воспользоваться функцией cr_pkbase_find().

4.6. Удаление открытого ключа ЭП из справочника

int DECL **cr_pkbase_remove(H_PKBASE** *pkbase_handle,*
const char **userid***);*

Назначение:

Функция удаляет открытый ключ пользователя из справочника.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, характеризующий справочник открытых ключей
userid	in	указатель на строку, содержащую идентификатор пользователя владельца открытого ключа, который должен быть удален из справочника

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_WRITE_FILE	Ошибка записи файла
ERR_MEM	Недостаточно памяти
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция определяет наличие в справочнике, характеризуемом параметром pkbase_handle, открытого ключа с идентификатором userid и в случае обнаружения удаляет его из справочника.

4.7. Сохранение справочника

int DECL **cr_pkbase_save**(**H_PKBASE** *pkbase_handle*
const char **pk_file* **);**

Назначение:

Функция сохраняет справочник открытых ключей в файле.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, характеризующий справочник открытых ключей
pk_file	in	имя файла, в который записывается справочник открытых ключей

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_PUB	Файла БОК не может быть открыт
ERR_MEM	Недостаточно памяти
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Данная функция может быть вызвана только в том случае, если база открытых ключей была открыта с использованием параметра flag_modify равным 1. Иначе выдается ошибка ERR_BAD_PARAM.
Функция сохраняет в файле, определяемом параметром pk_file, справочник открытых ключей.
После сохранения справочника в файле должен быть обеспечен контроль его целостности.
Для справочника с открытыми ключами пользователей контроль его целостности обеспечивается путем формирования ЭП с использованием ключей Администратора безопасности.
Для справочника с открытыми ключами Администраторов безопасности контроль его целостности обеспечивается внешней системой защиты от НСД (например, утилитой CheckHash.exe).

4.8. Выбор первого ключа из справочника открытых ключей

int DECL **cr_pkbase_findfirst** (**H_PKBASE** *pkbase_handle*,
H_PKEY **pkey_handle*,
void **comment*,
int **com_blen*);

Назначение:

Функция возвращает первый открытый ключ из справочника.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, возвращенный функцией cr_pkbase_open().
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ
comment	out	указатель на буфер, в который возвращается комментарий к открытому ключу
com_blen	in/out	на входе содержит максимальную длину буфера comment, на выходе – реальную возвращенную длину буфера comment

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция возвращает первый открытый ключ из справочника открытых ключей и комментарий к нему. Для выборки последующих ключей используйте функцию cr_pkbase_findnext(). Во время выборки, если вы еще не закончили выборку всех ключей, не пользуйтесь функцией cr_pkbase_find(), так как cr_pkbase_find() и cr_pkbase_findnext() работают с одними и теми же внутренними переменными дескриптора справочника открытых ключей.

4.9. Выбор следующего ключа из справочника открытых ключей

int DECL **cr_pkbase_findnext**(**H_PKBASE** *pkbase_handle*,
H_PKEY **pkey_handle*,
void **comment*,
int **com_blen*);

Назначение:

Функция возвращает очередной открытый ключ из справочника.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, возвращенный функцией cr_pkbase_open().
pkey_handle	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ
comment	out	указатель на буфер, в который возвращается комментарий к открытому ключу
com_blen	in/out	на входе содержит максимальную длину буфера comment, на выходе – реальную возвращенную длину буфера comment

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция возвращает очередной открытый ключ из справочника открытых ключей и комментарий к нему. Для выборки первого ключа используйте функцию cr_pkbase_findfirst(). Во время выборки, если вы еще не закончили выборку всех ключей, не пользуйтесь функцией cr_pkbase_find(), так как cr_pkbase_find() и cr_pkbase_findfirst() работают с одними и теми же внутренними переменными дескриптора справочника открытых ключей.

4.10. Закрытие справочника

int DECL **cr_pkbase_close(H_PKBASE** *pkbase_handle***);**

Назначение:

Функция освобождает дескриптор и закрывает файл, в котором содержится справочник открытых ключей.

Параметры:

pkbase_handle	in	дескриптор H_PKBASE, характеризующий справочник открытых ключей

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Функция закрывает дескриптор pkbase_handle, характеризующий справочник открытых ключей, и освобождает выделенную под него память и закрывает файл со справочником. Если в дальнейшем попытаться воспользоваться этим дескриптором, то будет выдаваться ошибка ERR_BAD_HANDLE.

4.11. Запись в ТМ-идентификатор 32-байтного значения

int DECL **cr_write_tm** (**void** **buf*,
int *buf_len*);

Назначение:

Функция записывает в ТМ-идентификатор 32 байта данных.

Параметры:

buf	in	указатель на буфер, в котором содержится записываемое значение
buf_len	in	длина данных в буфере buf (всегда должно быть 32 байта)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора

Описание функции:

Функция может быть использована для контроля целостности файлов (например, справочника с открытыми ключами администраторов). Значение хеш-функции для файла следует вычислить с помощью функции cr_hash_calc() и затем записать его в ТМ с помощью данной функции.
Запись данных с помощью данной функции может быть произведена только в ТМ типа Dallas1993 (тип ТМ номер 06), с размером внутренней памяти более 256 байт, в противном случае будет возвращен код ошибки ERR_READ_TM.
Тип ТМ, предъявленной пользователем, может быть определен с помощью функции cr_get_tm_number().
typedef struct {

```
    unsigned char FamilyCode; /* код типа ТМ */
    unsigned char SerNum[6];  /* серийный номер ТМ */
    unsigned char CRC;        /* контрольная сумма */
} TmIdRec;
```

Замечание 1. Значение, вычисленное с помощью функции cr_hash_calc(), имеет размер 32 байта. Из них в ТМ будут записаны только первые 30 байт, затем два байта контрольной суммы (CRC) вычисленной по этим 30 байтам. Всего будет записано 32 байта по смещению 256 в памяти ТМ-идентификатора.
Для чтения значения хеш-функции из ТМ следует использовать функцию cr_read_tm().

4.12. Чтение из ТМ-идентификатора 32-байтного значения

int DECL **cr_read_tm** (**void** **buf*,
int **buf_len*);

Назначение:

Функция считывает из ТМ-идентификатора 32-байтное значение данных.

Параметры:

buf	in	указатель на буфер, в который возвращается значение
buf_len	in/out	на входе содержит максимальную длину буфера buf, на выходе - реальная длина данных, записанных в буфер buf (всегда должно быть 32 байта)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора

Описание функции:

Функция может быть использована для контроля целостности файлов (например, справочника с открытыми ключами администраторов). Текущее значение для файла следует вычислить с помощью функций cr_hash_calc() и затем сравнить его с эталонным значением, прочитанным из ТМ с помощью данной функции.
Чтение данных с помощью данной функции может быть произведено только из ТМ типа Dallas1993 (тип ТМ номер 06), в противном случае будет возвращен код ошибки ERR_READ_TM.
Тип ТМ, предъявленной пользователем, может быть определен с помощью функции cr_get_tm_number().
Замечание 1. Значение, вычисленное с помощью функций cr_hash_calc(), имеет размер 32 байта. Для сравнения со значением, прочитанным из ТМ, следует использовать только первые 30, т.к. в двух последних байтах содержится CRC.
Для записи значения хеш-функции в ТМ следует использовать функцию cr_write_tm().

4.13. Чтение трех страниц из ТМ-идентификатора

int DECL **cr_read_three_tm_page** (**H_INIT** *init_handle*,
void **pagebufer*);

Назначение:

Функция считывает из ТМ-идентификатора значение трех страниц памяти. Используется для резервного сохранения (backup) информации из ТМ-идентификатора.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
pagebufer	out	указатель на буфер, в который возвращается содержимое ТМ, размер 96 байт

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV

Описание функции:

Используется для резервного хранения информации, записанной на ТМ-носителе.

4.14. Установка в контекст открытого ключа

int DECL **cr_pkey_put** (**void** **pubkey*,
char **namkey*,
H_PKEY **struct_pkey*);

Назначение:

Функция импортирует из буфера открытый ключ и создает дескриптор, характеризующий открытый ключ.

Параметры:

pubkey	in	указатель на буфер, в котором содержится открытый ключ в формате Бикрипт (256 байт)
namkey	in	указатель на идентификатор открытого ключа (не более 32-х байт, заканчивающихся нулем)

struct_pkey	out	возвращаемый дескриптор H_PKEY, характеризующий открытый ключ, который можно использовать в других функциях

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция используется для импорта открытого ключа из буфера и для создания дескриптора H_PKEY, характеризующего открытый ключ, который может использоваться в других функциях.

4.15. Загрузка открытого ключа из области памяти

int DECL cr_pkey_load (**H_INIT** *Ainit_structX,*
**void ***Apubkey,*
int *pubkeyLength,*
**char ***namkey,*
char *Aparam,*
**H_PKEY ***Apkey_struct);*

Назначение:

Функция импортирует из буфера открытый ключ “в сыром виде” и создает дескриптор, характеризующий открытый ключ.

Параметры:

Ainit_structX	in	Дескриптор библиотеки
Apubkey	in	Буфер открытого ключа
pubkeyLength	in	Длина буфера открытого ключа
namkey	in	Идентификатор пользователя ключа
Aparam	in	Параметр эллиптической кривой ключа (см. приложение №3)
Apkey_struct	out	Дескриптор открытого ключа

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MEM	Недостаточно памяти
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция используется для импорта открытого ключа из буфера и для создания дескриптора H_PKEY, характеризующего открытый ключ, который может использоваться в других функциях.

4.16. Получение идентификатора открытого ключа

int DECL **cr_pkey_getid** (**H_PKEY** *pkey_handle*,
char **userid*,
int **userid_blen*);

Назначение:

Функция экспортирует в буфер идентификатор открытого ключа.

Параметры:

pkey_handle	in	дескриптор H_PKEY, характеризующий открытый ключ
userid	out	указатель на буфер, в который возвращается идентификатор открытого ключа
userid_blen	in/out	на входе содержит максимальную длину буфера userid, на выходе – реальную возвращенную длину буфера userid

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция возвращает идентификатор открытого ключа, характеризуемый дескриптором H_PKEY.

4.17. Экспортирование открытого ключа

int DECL **cr_pkey_getinfo** (**H_PKEY** *pkey_handle*,
char **userid*,
int **userid_blen*,
void **pkbuf,*
int **pkbuf_blen*);

Назначение:

Функция экспортирует открытый ключ (например, для распечатки на бумаге).

Параметры:

pkey_handle	in	дескриптор H_PKEY, характеризующий открытый ключ
userid	out	указатель на буфер, в который возвращается идентификатор, не более 32-х байт
userid_blen	in/out	на входе содержит максимальную длину буфера userid, на выходе – реальную возвращенную длину буфера userid
pkbuf	out	указатель на буфер, в который возвращается открытый ключ, размер 256 байт
pkbuf_blen	in/out	на входе содержит максимальную длину буфера pkbuf, на выходе – реальную возвращенную длину буфера pkbuf

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция возвращает открытый ключ и идентификатор открытого ключа в буфер. Причем возвращается открытый ключ размером 256 байт. Значимыми являются только первые 32 байта ключа и еще 32 байта, начинающиеся со смещения (offset) 64.

4.18. Деинициализация открытого ключа

int DECL **cr_pkey_close(H_PKEY** *pkey_handle* **);**

Назначение:

Функция закрывает дескриптор H_PKEY, характеризующий открытый ключ.

Параметры:

pkey_handle	in	дескриптор H_PKEY, характеризующий открытый ключ пользователя

Возвращаемое значение:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована через cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Функция закрывает дескриптор H_PKEY. Дальнейшая работа с дескриптором невозможна и ведет к возврату ошибки ERR_BAD_HANDLE.

4.19. Получение информации о средствах ЭП сертификата X.509

int DECL **cr_get_cert_usage_info** (**H_INIT** *init_handle*,
char **certificateBuffer*,
int *certificateLength,*
char **certificateStr111*,
int **certificateLength111,*
char **certificateStr112*,
int **certificateLength112,*
char **certificateStr113*,
int **certificateLength113*);

Назначение:

Функция возвращает информацию о средствах ЭП и разрешенных применениях ключа, указанных в сертификате.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
certificateBuffer	in	буфер с загруженным сертификатом в формате DER
certificateLength		длина буфера с загруженным сертификатом в формате DER
certificateStr111	out	указатель на буфер, в который возвращается информация oid = 1.2.643.100.111 “Описание средства ЭП владельца сертификата” В информации может быть несколько строк, заканчивающихся нулем. Конец информации – всегда двойной нуль.
certificateLength111	in/out	на входе содержит максимальную длину буфера certificateStr111, на выходе – реальную возвращенную длину буфера certificateStr111

certificateStr112	out	указатель на буфер, в который возвращается информация oid = 1.2.643.100.112 “Описание средства ЭП издателя сертификата” В информации может быть несколько строк, заканчивающихся нулем. Конец информации – всегда двойной нуль.
certificateLength112	in/out	на входе содержит максимальную длину буфера certificateStr112, на выходе – реальную возвращенную длину буфера certificateStr112
certificateStr113	out	указатель на буфер, в который возвращается информация oid = 2.5.29.32 “Политики сертификата” В информации может быть несколько строк, заканчивающихся нулем. Конец информации – всегда двойной нуль.
certificateLength113	in/out	на входе содержит максимальную длину буфера certificateStr113, на выходе – реальную возвращенную длину буфера certificateStr113

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_LONG_LENGTH	Длина закодированного OID имеет размер более 4 байт

Описание функции:

Функция возвращает информацию о средствах ЭП для сертификата открытого ключа в формате DER стандарта X.509.

5 Функции формирования и проверки ЭП

5.1. Формирование ЭП для блока памяти

int DECL **cr_sign_buf(H_INIT** *init_handle*,
H_USER *user_handle*,
**void ***buf*,
int *buf_len*,
**void ***sign*,
**int ***sign_blen*);

Назначение:

Функция вычисляет ЭП для блока информации, размещенной в буфере.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey(), cr_load_elgkey_mk()
Buf	in	указатель на буфер, для которого должна быть сформирована электронная цифровая подпись
buf_len	in	длина буфера buf
Sign	out	указатель на буфер, в который возвращается ЭП для буфера buf
sign_blen	in/out	на входе содержит максимальную длину буфера sign, на выходе - реальная длина сформированной структуры ЭП, возвращаемой через буфер sign (64 байта)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_SIGN	ЭП не верна
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_MEM	Недостаточно памяти
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Функция формирует ЭП для информационного буфера buf длины buf_len, помещает ЭП в буфер sign.
Буфер buf может содержать как текстовую, так и бинарную информацию.
Функция доступна, если в результате вызова функции cr_init() установлен флаг cDSCH.
Если размер буфера sign недостаточен, функция возвращает код ошибки ERR_MORE_DATA, через параметр sign_blen возвращается необходимый размер буфера.

5.2. Проверка ЭП для блока памяти

int DECL **cr_check_buf(H_INIT** *init_handle,*
H_PKEY *pkey_handle*,
void **buf*,
int *buf_blen*,
**void ***sign*,
int *sign_blen*);

Назначение:

Функция проверяет правильность ЭП для блока памяти.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
pkey_handle	in	дескриптор H_PKEY, возвращенный функцией cr_pkbase_find() или cr_pkey_put() и характеризующий открытый ключ, с помощью которого должна быть проверена ЭП
buf	in	указатель на буфер, для которого должна быть проверена правильность ЭП
buf_len	in	длина буфера buf
sign	in	указатель на буфер, в котором содержится ЭП
sign_blen	in	размер буфера sign (64 байта)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCO RE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция проверяет ЭП для буфера данных buf длины buf_blen, используя открытый ключ pkey_handle.
Буфер buf может содержать как текстовую, так и бинарную информацию.
Если проверка правильности ЭП с заданным открытым ключом дала отрицательный результат, функция возвращает код ошибки ERR_BAD_SIGN.
Буфер подписи buf может быть извлечен из подписанного буфера функцией cr_buf_get_sign_struct().

Дескриптор открытого ключа pkey_handle может быть получен при помощи функции cr_pkbase_find(). Функция доступна всегда.

5.3. Формирование ЭП для значения хеш-функции

int DECL **cr_sign_hash(H_INIT** *init_handle*,
H_USER *user_handle*,
void * *hash*,
int *hash_len*,
void * *sign*,
int * *sign_blen*);

Назначение:

Функция вычисляет ЭП для значения хеш-функции.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey(), cr_load_elgkey_mk(), cr_read_skey(), cr_read_skey_mk()
hash	in	буфер, в котором содержится значение хеш-функции
hash_len	in	длина буфера hash
sign	out	указатель на буфер, в который возвращается ЭП для значения hash
sign_blen	in/out	на входе содержит максимальную длину буфера sign, на выходе - реальная длина сформированной структуры ЭП, возвращаемой через буфер sign

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_SIGN	ЭП не верна
ERR_READ_TM	Ошибка чтения ТМ-идентификатора

ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Функция формирует ЭП для значения хеш-функции hash длины hash_blen, помещает ЭП в буфер sign.
Буфер hash может содержать как текстовую, так и бинарную информацию.
Функция доступна, если в результате вызова функции cr_init() был установлен флаг cDSCH.
Если размер буфера sign недостаточен, функция возвращает код ошибки ERR_MORE_DATA, через параметр sign_blen возвращается необходимый размер буфера.

Примечание:

При вызове этой функции из прикладного программного обеспечения, необходимо проведение дополнительных тематических исследований.

5.4. Проверка ЭП для значения хеш-функции

int DECL **cr_check_hash(H_INIT** *init_handle,*
H_PKEY *pkey_handle***,**
void **hash***,*
int *hash_len***,**
void ***sign***,**
int *sign_len***);**

Назначение:

Функция проверяет правильность ЭП для значения хеш-функции.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
pkey_handle	in	дескриптор H_PKEY, возвращенный функцией cr_pkbase_find() и характеризующий открытый ключ, с помощью которого должна быть проверена ЭП
hash	in	указатель на буфер, в котором содержится значение хеш-функции, для которого должна быть проверена правильность электронной подписи
hash_len	in	длина буфера hash
sign	in	указатель на буфер, в котором содержится структура ЭП
sign_len	in	размер буфера sign

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее

ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCO RE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Функция проверяет ЭП для буфера данных hash длины hash_len, используя открытый ключ pkey_handle.
Буфер hash должен иметь размер 32 байта и содержать бинарную информацию.
Если проверка правильности ЭП с заданным открытым ключом дала отрицательный результат, функция возвращает код ошибки ERR_BAD_SIGN.
Функция доступна всегда.

Примечание:

При вызове этой функции из прикладного программного обеспечения, необходимо проведение дополнительных тематических исследований.

5.5. Формирование ЭП для содержимого файла

int DECL **cr_sign_file(H_INIT** *init_handle***,**
H_USER *user_handle***,**
const char * *file_name***);**

Назначение:

Функция вычисляет ЭП для содержимого файла и записывает полученную ЭП в конец этого файла.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey(), cr_load_elgkey_mk()
file_name	in	полный путь и имя подписываемого файла

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее

ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_MEM	Недостаточно памяти
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_READ_FILE	Ошибка чтения файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_SIGN	ЭП не верна
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция формирует ЭП для значения содержимого файла file_name и записывает контрольную сумму (2 байта), ЭП (64 байта), идентификатор пользователя - владельца открытого ключа ЭП (от 1 до 32 байта) и длину этого идентификатора (1 байт) в конец этого файла. Далее записываются 4 символа "BICS".
Файл может содержать как текстовую, так и бинарную информацию.
Подпись в последующем может быть проверена и удалена функцией cr_check_file().
Функция может быть вызвана несколько раз для одного и того же файла, в результате файл может быть подписан несколько раз разными пользователями. При подписи осуществляется сцепление подписей, то есть каждая последующая подпись подписывает как сам файл, так и все предыдущие подписи.
Функция доступна, если в результате вызова функции cr_init() был установлен флаг cDSCH.

5.6. Формирование ЭП для содержимого файла, открытого приложением

int DECL **cr_sign_hfile(H_INIT** *init_handle*,
H_USER *user_handle*,
HANDLE *hFile*);

Назначение:

Функция вычисляет ЭП для содержимого файла и записывает полученную ЭП в конец этого файла.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
user_handle	in	дескриптор H_USER, возвращенный функцией cr_load_elgkey(), cr_load_elgkey_mk()

hFile	in	дескриптор подписываемого файла, результат вызова функции Windows API CreateFile() с параметрами GENERIC_READ | GENERIC_WRITE (доступ на чтение и на запись)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_MEM	Недостаточно памяти
ERR_READ_FILE	Ошибка чтения файла
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_SIGN	ЭП не верна
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_LOAD_GRN_DLL	Ошибка загрузки DLL
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция формирует ЭП для значения содержимого файла hFile, открытого приложением. Файл должен быть открыт с доступом на запись. Функция записывает контрольную сумму (2 байта), ЭП (64 байта), идентификатор пользователя - владельца открытого ключа ЭП (от 1 до 32 байта) и длину этого идентификатора (1 байт) в конец этого файла. Далее записываются 4 символа "BICS".
Файл может содержать как текстовую, так и бинарную информацию.
Подпись в последующем может быть проверена и удалена функцией cr_check_file().
Функция может быть вызвана несколько раз для одного и того же файла, в результате файл может быть подписан несколько раз разными пользователями. При подписи осуществляется сцепление подписей, то есть каждая последующая подпись подписывает как сам файл, так и все предыдущие подписи.
Функция доступна, если в результате вызова функции cr_init() был установлен флаг cDSCH.

Функция доступна только в среде ОС семейства Windows.

5.7. Проверка ЭП для содержимого файла

int DECL **cr_check_file(H_INIT** *init_handle,*
H_PKBASE *pkbase_handle*,
const char **file_name*,
int *N*,
int *flag_del*,
char ***userid*,
int ***userid_blen*);

Назначение:

Функция проверяет правильность подписи номер N для содержимого файла и, если установлен флажок flag_del, удаляет ее из файла.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
pkbase_handle	in	дескриптор H_PKBASE, возвращенный функцией cr_pkbase_open() и характеризующий базу открытых ключей, с помощью которой должна быть проверена ЭП
file_name	in	имя файла, для которого должна быть проверена правильность электронной подписи
N	in	номер подписи, которая должна быть проверена: 1 - последняя, 2 - предпоследняя и т.д.
flag_del	in	если равен 1, то в случае, если подпись есть (контрольная сумма совпала), она будет удалена из файла если равен 0, подпись не удаляется при N равным 2 и более будут удалены все подписи от 1 до N. После удаления подписи и последующих вызовах функции для проверки одного и того же файла N необходимо снова устанавливать равным 1.
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца подписи
userid_blen	in/out	на входе максимальный размер буфера userid, на выходе - реальный возвращенный размер буфера userid

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст

ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_WRITE_FILE	Ошибка записи файла
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_READ_FILE	Ошибка чтения файла
ERR_SIGN_NO_REG	ЭП не зарегистрирована в файле БОК
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCO RE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Функция проверяет существование подписи (для этого должна совпасть контрольная сумма) под номером N для содержимого файла. Если подписи нет, возвращается ошибка ERR_NO_SIGN.
Если подпись есть (не обязательно это действительно ЭП, главное, чтобы на блок совпала контрольная сумма) и если параметр flag_del=1, подпись будет проверена и удалена из файла (не обязательно правильная).
Функция проверяет ЭП, используя открытый ключ, идентификатор которого извлекается из проверяемого файла file_name, затем осуществляется поиск открытого ключа с этим идентификатором в справочнике pkbase_handle.
Если открытый ключ не найден, возвращается ошибка ERR_SIGN_NO_REG.
Если ключ найден, но подпись не верна, возвращается ошибка ERR_BAD_SIGN.
Идентификатор пользователя возвращается независимо от того, верна подпись или неверна.
Функция доступна всегда.

5.8. Помещение структуры данных подписи в конец буфера

int DECL **cr_buf_put_sign_struct (H_INIT** *init_handle,*
void **buf*,
int *inbuf_len*,
**int ***outbuf_len*,
**void ***sign*,
int *sign_blen*,
**char ***userid*,
int *userid_blen*);

Назначение:

Функция создает контрольную сумму на блок подписи и помещает структуру блока подписи в конец буфера.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
buf	in	указатель на буфер, заполненный данными, в конце которого, после вызова функции, должна быть помещена структура подписи

inbuf_len	in	размер этого буфера, заполненного данными, в конце которого, после вызова функции, должна быть помещена структура подписи
outbuf_len	in/out	на входе – максимальный размер буфера, на выходе – реальный размер буфера после записи в него структуры подписи
sign	in	указатель буфер, в котором содержится подпись, которая будет помещена в структуру подписи
sign_blen	in	размер буфера sign
userid	in	указатель буфер, в котором содержится идентификатор пользователя владельца подписи, который будет помещен в структуру подписи
userid_blen	in	размер буфера userid

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция создает контрольную сумму на блок подписи и помещает структуру блока подписи в буфер по адресу inbuf_len.
Дополнительная длина буфера (outbuf_len – inbuf_len) должна быть не меньше, чем 2 (контрольная сумма) + <размер подписи> + <размер идентификатора> + 1 (длина идентификатора) + 4 (строка "BICS"). Иначе выдается ошибка ERR_MORE_DATA и в переменную outbuf_len заносится необходимый размер.
Функция доступна всегда.

5.9. Помещение структуры данных подписи в конец файла

int DECL **cr_file_put_sign_struct(H_INIT** *init_handle,*
const char **file_name*,
void ***sign,**
int *sign_blen*,
char ***userid,**
int *userid_blen*);

Назначение:

Функция создает контрольную сумму на блок подписи и помещает структуру блока подписи в конец файла с именем file_name.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
file_name	in	имя файла, в который должна быть помещена структура подписи
sign	in	указатель буфер, в котором содержится подпись, которая должна быть помещена в файл
sign_blen	in	размер буфера sign
userid	in	указатель буфер, в котором содержится идентификатор пользователя владельца подписи, который должен быть помещен в файл
userid_blen	in	размер буфера userid

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Функция создает структуру блока подписи (контрольная сумма (2 байта), ЭП (64 байта), идентификатор пользователя - владельца открытого ключа ЭП (от 1 до 32 байта) и длина этого идентификатора (1 байт)) и помещает ее в конец файла с именем file_name. Далее записываются 4 символа "BICS". Файл должен быть доступен на запись.

5.10. Получение структуры данных подписи из буфера

int DECL **cr_buf_get_sign_struct (H_INIT** *init_handle,*
void **buf*,
int *buf_len*,
int *search_from*,
**void ***sign*,
**int ***sign_blen*,
**char ***userid*,
**int ***userid_blen*,
**int ***struct_blen*);

Назначение:

Функция проверяет, используя подсчет контрольной суммы (CRC), есть ли структура подписи в конце буфера и если есть, возвращает эту структуру. Подпись из буфера не удаляется.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
buf	in	указатель на буфер, в котором содержатся данные и структура подписи

buf_len	in	размер этого буфера
search_from	in	если несколько подписей, то это расстояние от конца буфера до подписи которую нужно вернуть (расстояние в байтах) для первой (последней с конца) подписи должно быть равно нулю
sign	out	указатель на буфер, в который возвращается подпись
sign_blen	in/out	на входе максимальный размер буфера sign, на выходе - реальный возвращенный размер буфера sign
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца подписи
userid_blen	in/out	на входе максимальный размер буфера userid, на выходе - реальный возвращенный размер буфера userid
struct_blen	out	реальный размер структуры подписи внутри файла, следующая подпись (если она есть) будет размещаться на расстоянии search_from + struct_blen от конца файла

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция проверяет существование подписи (для этого должна совпасть контрольная сумма) на расстоянии search_from от конца буфера. Если подписи нет, возвращается ошибка ERR_NO_SIGN.
Если подпись есть (не обязательно, что это будет действительно ЭП, возможно, что на блок совпала контрольная сумма и 4 символа "BICS" в конце), то сама подпись и идентификатор пользователя владельца подписи возвращаются в буферах sign и userid.
Полная длина структуры подписи возвращается в переменной struct_blen. Для удаления подписи следует уменьшить размер буфера на search_from + struct_blen байт.
Подпись и идентификатор пользователя возвращается независимо от того, верна подпись или неверна, главное чтобы совпала контрольная сумма на блок подписи внутри буфера.
Для нескольких подписей можно организовать следующий цикл:

- 1-ая подпись:
  устанавливаем
  `search_from = 0;`
  и вызываем
  `cr_buf_get_sign_struct();`
- 2-ая подпись (и последующие, до тех пор, пока не возникнет ERR_NO_SIGN):
  увеличиваем
  `search_from = search_from + struct_blen;`
  где значение struct_blen - вернул предыдущий вызов cr_buf_get_sign_struct()
  и вызываем
  `cr_buf_get_sign_struct();`

Функция доступна всегда.

5.11. Получение структуры данных подписи из файла

int DECL **cr_file_get_sign_struct(H_INIT** *init_handle,*
const char **file_name*,
int *search_from*,
void ***sign*,
int ***sign_blen*,
char ***userid*,
int ***userid_blen*,
int ***struct_blen*);

Назначение:

Функция проверяет есть ли структура подписи в конце файла с именем file_name и, если есть, возвращает эту структуру. Подпись из файла не удаляется.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
file_name	in	имя файла, из которого будет прочитана структура подписи
search_from	in	если несколько подписей, то это расстояние от конца файла до подписи которую нужно вернуть (расстояние в байтах) для первой (последней с конца) подписи должно быть равно нулю
sign	out	указатель на буфер, в который возвращается подпись
sign_blen	in/out	на входе максимальный размер буфера sign, на выходе - реальный возвращенный размер буфера sign
userid	out	указатель на буфер, в который возвращается идентификатор пользователя владельца подписи
userid_blen	in/out	на входе максимальный размер буфера userid, на выходе - реальный возвращенный размер буфера userid
struct_blen	out	реальный размер структуры подписи внутри файла, следующая подпись (если она есть) будет размещаться на расстоянии search_from + struct_blen от конца файла

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_READ_FILE	Ошибка чтения файла
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Функция проверяет существование подписи (для этого должна совпасть контрольная сумма) на расстоянии search_from от конца файла. Если подписи нет, возвращается ошибка ERR_NO_SIGN.
Если подпись есть (не обязательно, что это будет действительно ЭП, возможно, что на блок совпала контрольная сумма и 4 символа "BICS" в конце), то сама подпись и идентификатор пользователя - владельца подписи возвращаются в буферах sign и userid.
Полная длина структуры подписи возвращается в переменной struct_blen. Для удаления подписи следует уменьшить размер файла на search_from + struct_blen байт.
Подпись и идентификатор пользователя возвращается независимо от того, верна подпись или неверна, главное чтобы совпала контрольная сумма на блок подписи внутри файла.
Для нескольких подписей можно организовать следующий цикл:

- 1-ая подпись:
  устанавливаем
  search_from = 0;
  и вызываем
  cr_file_get_sign_struct();
- 2-ая подпись (и последующие, до тех пор, пока не возникнет ERR_NO_SIGN):
  увеличиваем
  search_from = search_from + struct_blen;
  где значение struct_blen - вернул предыдущий вызов cr_ file_get_sign_struct()
  и вызываем
  cr_ file_get_sign_struct();

Функция доступна всегда.

6 Функции ДСЧ

6.1. Генерация случайной последовательности

int DECL **cr_get32_random (H_INIT** *init_handle,*
char **data*);**

Назначение:

Заполняет буфер размером 32 байта случайными числами.

Параметры:

init_handle	in	дескриптор H_INIT, возвращенный функцией cr_init().
Data	out	указатель на буфер, размером 32 байта, в который возвращается значение случайных чисел

Возвращаемое значение:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки

ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования

Описание функции:

Для заполнения буфера используется программный датчик случайных чисел.

7 Функции верхнего уровня

7.1. Служебные функции

7.1.1. *Получение следующего индекса цепочки сертификатов*

H_STORAGE DECL cr_GetNextByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN);*

Назначение:

Возвращает следующий индекс цепочки сертификатов или BAD_H_STORAGE если следующего элемента в цепочке нет.

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

Описание функции:

Функции проверки ЭП возвращают первый элемент цепочки сертификатов, сертификат подписанта. Для получения следующего элемента (индекса в хранилище) следует использовать данную функцию.

7.1.2. *Получение корневого сертификата в цепочке сертификатов*

H_STORAGE DECL cr_GetRootByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN);*

Назначение:

Получает корневой сертификат в цепочке сертификатов.

Параметры:

ctx	in	Контекст библиотеки

itemN	in	Индекс сертификата в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

Описание функции:

Функции проверки ЭП возвращают первый элемент цепочки сертификатов, сертификат подписанта. Для получения индекса корневого сертификата следует использовать данную функцию. Если цепочка не имеет корневого сертификата, например, при построении возникла ошибка, данная функция вернет BAD_H_STORAGE.

7.1.3. ***Получение имени файла сертификата в хранилище***

Int32 DECL cr_GetFilenameByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN,*
char ** *buffer,*
UInt32 ** *bufferLength);*

Назначение:

Получает имя файла сертификата в хранилище

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище
buffer	out	Буфер с именем файла в кодировке Win-1251, с результирующим нулем
bufferLength	in/out	Вход - максимально возможный размер буфера с именем файла, выход - длина буфера с именем файла

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Если сертификат был загружен в хранилище из файла, данная функция вернет имя файла и полный путь. Если сертификат был загружен из CMS, функция вернет пустой буфер.

7.1.4. ***Получение сертификата из хранилища***

Int32 DECL cr_GetBufferByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN,*
UInt8* *buffer,*
UInt32* *bufferLength);*

Назначение:

Получает закодированный в кодировке ASN.1 сертификат из хранилища.

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище
buffer	out	Буфер сертификата
bufferLength	in/out	Вход - максимально возможный размер буфера сертификата, выход - длина буфера сертификата

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Позволяет извлечь сертификат из хранилища по его индексу.

7.1.5. ***Получение статуса сертификата из хранилища***

Int32 DECL cr_GetStatusByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN,*
Int32 * *Astatus);*

Назначение:

Получает статус сертификата из хранилища

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище
Astatus	out	Статус сертификата, 0-успешно проверен, BAD_STATUS-не проверялся или код ошибки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CERTIFICATE_ NOT_FOUND	Не найден следующий сертификат в цепочке

Описание функции:

Позволяет получить статус сертификата из хранилища по его индексу. Статус 0 - сертификат успешно проверен. Статус BAD_STATUS - сертификат не проверялся. Статус отличный от 0 и BAD_STATUS - сертификат был проверен по хранилищу с ошибкой. Статус равен коду ошибки (см. Приложение 1). При добавлении сертификатов в хранилище путем вызова функции cr_AddCertificate все сертификаты получают статус BAD_STATUS. Для проверки ЭП загруженного в хранилище сертификата используйте cr_CheckCertificateByStorage, которая изменит статус сертификата. После добавления новых сертификатов в хранилище, статус ранее добавленных сертификатов не меняется. Для проверки ЭП загруженных ранее сертификатов, в том числе имеющих статус, отличный от BAD_STATUS используйте функцию cr_CheckCertificateByStorage. При формировании ЭП происходит контроль времени действия сертификатов и списков отозванных сертификатов, участвующих в данной процедуре (сертификаты из цепочки подписанта и СОС, если таковые были загружены в контекст). Все они должны быть действительны на текущий момент времени. Иначе метод SignInit вернет соответствующую ошибку. При этом если ни один СОС не был загружен в контекст, это не является ошибкой. В таком случае будет осуществляться контроль действия только сертификатов из цепочки подписанта.

7.1.6. ***Получение длины цепочки сертификатов в хранилище***

Int32 DECL cr_GetChainSizeByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN);*

Назначение:

Получает длину цепочки сертификатов в хранилище

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс первого сертификата в цепочке

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

Описание функции:

Функции проверки ЭП возвращают первый элемент цепочки сертификатов, для получения длины цепочки используйте функцию cr_GetChainSizeByStorage

7.1.7. *Получение количества сертификатов и списка отозванных сертификатов в хранилище*

Int32 DECL cr_GetStorageInfo(**H_CTX** *ctx,*
UInt32 * *numof_cert,*
UInt32 * *numof_crl);*

Назначение:

Получает количество сертификатов и количество списков отозванных сертификатов в хранилище. Стоп-листы могут попасть в хранилище только после проверки по сроку действия, проверки ЭП стоп-листа и успешному построению цепочки сертификатов подписанта, с проверкой срока действия сертификатов в цепочке и окончанием цепочки корневым сертификатом, находящимся в конфигурационном файле в списке корневых.

Параметры:

ctx	in	Контекст библиотеки
numof_cert	out	Количество сертификатов в хранилище
numof_crl	out	Количество проверенных стоп-листов в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Если необходимо строго учитывать наличие стоп-листов, следует проверять, что в хранилище находится хотя бы один стоп-лист и прекращать работу с ошибкой если нет ни одного стоп-листа.

7.1.8. *Создание контекста времени*

Int32 DECL cr_CreateTime(**H_TIMER *** *timer);*

Назначение:

Создает контекст времени, в котором проходят криптографические операции.

Параметры:

timer	out	Контекст времени

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Возвращает контекст времени, который, после использования, следует освободить через cr_FreeTime.

7.1.9. *Освобождение контекста времени*

Int32 DECL cr_FreeTime(**H_TIMER** *timer);*

Назначение:

Освобождает контекст времени.

Параметры:

timer	in	Контекст времени для освобождения

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Освобождает контекст времени после использования. Если не освободить контекст времени, функция cr_Finalize вернет код ошибки 1001. Если не освободить два контекста времени, функция cr_Finalize вернет код ошибки 1002 и т.д.

7.1.10. *Установка текущего времени для контекста времени*

Int32 DECL cr_SetCurrentTime(**H_TIMER** *timer);*

Назначение:

Устанавливает текущее системное время для контекста времени.

Параметры:

timer	out	Заполненный контекст времени

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_INVALID_TIME	Неверное время

Описание функции:

В функции проверки ЭП cr_CheckInit можно передавать нулевой контекст времени (NULL) это эквивалентно проверке ЭП для момента текущего системного времени.

7.1.11. *Установка заданного времени в контекст*

Int32 DECL cr_SetTime(**H_TIMER** *timer,*
UInt32 *d,*
UInt32 *m,*
UInt32 *y,*
UInt32 *h,*
UInt32 *mm,*
UInt32 *s);*

Назначение:

Устанавливает заданное системное время в контекст.

Параметры:

timer	out	Заполняемый контекст времени

d	in	день 1-31
m	in	месяц 1-12
y	in	год 1900-2500
h	in	час 0-23
mm	in	минуты 0-59
s	in	секунды 0-59

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_INVALID_TIME	Неверное время

Описание функции:

В функции проверки ЭП cr_CheckInit можно задавать время, что приведет к проверке сертификатов и стоп-листов на соответствие срока действия заданному моменту времени.

7.1.12. *Получение времени из контекста*

Int32 DECL cr_GetTime(**H_TIMER** *timer*,
UInt32 * *d*,
UInt32 * *m*,
UInt32 * *y*,
UInt32 * *h*,
UInt32 * *mm*,
UInt32 * *s);*

Назначение:

Получение времени из контекста.

Параметры:

timer	in	Заполненный контекст времени
d	out	день 1-31
m	out	месяц 1-12
y	out	год 1900-2500
h	out	час 0-23
mm	out	минуты 0-59
s	out	секунды 0-59

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Позволяет получить время из контекста времени, например, срок действия сертификата.

7.1.13. *Копирование контекста времени*

Int32 DECL cr_CopyTime(**H_TIMER** *timer1,*
H_TIMER *timer2);*

Назначение:

Копирование времени в контексте времени.

Параметры:

timer1	in	Контекст времени источника
timer2	in	Контекст времени получателя

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Позволяет скопировать время из одного контекста времени в другой. Оба контекста должны существовать.

7.1.14. *Политика сертификата и класс средства ЭП*

Int32 DECL cr_GetCertificateUsageInfo(**H_CTX** *ctxX,*
UInt8 * *certBuffer,*
UInt32 *certBufferLength,*
char * *s1,*
int * *len_s1,*
char * *s2,*
int * *len_s2,*
char * *s3,*
int * *len_s3);*

Назначение:

Функция возвращает информацию о средствах ЭП и разрешенных применениях ключа, указанные в сертификате.

Параметры:

ctxX	in	Контекст библиотеки
certBuffer	in	Буфер сертификата в формате ASN.1 или PEM
certBufferLength	in	Длина буфера сертификата
s1	out указатель	на буфер, в который возвращается информация OID = 1.2.643.100.111 “Описание средства ЭП владельца сертификата”. В информации может быть несколько строк, заканчивающихся нулем. Конец информации - двойной нуль.
len_s1	in/out	На входе максимальная длина, на выходе длина строки описания средства ЭП владельца сертификата
s2	out	указатель на буфер, в который возвращается информация OID = 1.2.643.100.112 “Описание средства ЭП издателя сертификата”. В информации может быть несколько строк, заканчивающихся нулем. Конец информации - двойной нуль.
len_s2	in/out	На входе максимальная длина, на выходе длина строки описания средства ЭП издателя сертификата”
s3	out	указатель на буфер, в который возвращается информация OID = 2.5.29.32 “Политики сертификата”. В информации может быть несколько строк, заканчивающихся нулем. Конец информации - двойной нуль.
len_s3	in/out	На входе максимальная длина, на выходе длина строки политики сертификата

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер

Описание функции:

Функция возвращает информацию о средствах ЭП и разрешенных применениях ключа, указанные в сертификате.

7.1.15. *Удаление ЭП из CMS*

Int32 DECL cr_DeleteSignBuffer(**H_CTX** *ctxX,*
UInt32 *SignerNumber,*
UInt8 * *encodedCms,*
UInt32 *encodedCmsLength,*
UInt8 * *resultCms,*
UInt32 * *resultCmsLength);*

Назначение:

Удаление ЭП из CMS.

Параметры:

ctxX	in	Контекст библиотеки
SignerNumber	in	Номер удаляемой ЭП (нумерация начинается с 1)
encodedCms	in	Буфер CMS с удаляемыми ЭП в формате ASN.1 или PEM
encodedCmsLength	in	Длина буфера CMS с удаляемыми ЭП
resultCms	out	Буфер CMS с результирующей ЭП, в формате ASN.1 или PEM
resultCmsLength	in/out	Вход - максимально возможный размер буфера CMS с результирующей ЭП, выход - длина буфера CMS с результирующей ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_CHOICE_TYPE	Ошибка определения типа

Описание функции:

Удаляет ЭП с номером SignerNumber из CMS.

7.1.16. *Получение информации об ЭП из CMS в буфере*

Int32 DECL cr_GetCmsData(**H_CTX** *ctxX,*
**UInt8 *****encodedCms,*
UInt32 *encodedCmsLength,*
**UInt8 *****Data,*
**UInt32 *****DataLength,*
**Int32 *****Version,*
**UInt32 *****AlgorithmsCount,*
**UInt32 *****SignerInfoCount,*
**UInt32 *****AttachedCertificatesCount,*
**UInt32 *****CrlsCount);*

Назначение:

Получение информации об ЭП из CMS в буфере.

Параметры:

ctxX	in	Контекст библиотеки
encodedCms	in	Буфер CMS в формате ASN.1 или PEM
encodedCmsLength	in	Длина буфера CMS
Data	out	Буфер с подписанными данными (данные присутствуют в CMS только в случае присоединенной ЭП)
DataLength	in/out	Вход - максимально возможный размер буфера с подписанными данными, выход - длина буфера с подписанными данными
Version	out	Версия CMS
AlgorithmsCount	out	Количество алгоритмов хеширования в CMS
SignerInfoCount	out	Количество ЭП в CMS
AttachedCertificatesCount	out	Количество присоединенных сертификатов в CMS
CrlsCount	out	Количество присоединенных стоп-листов в CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1

ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Получает информации об ЭП с номером SignerNumber из CMS в буфере.

7.1.17. *Получение информации об ЭП из CMS в файле*

Int32 DECL cr_GetFileData(**H_CTX** *ctxX,*
const char * *currentCmsFile,*
UInt8 * *Data,*
UInt32 * *DataLength,*
Int32 * *Version,*
UInt32 * *AlgorithmsCount,*
UInt32 * *SignerInfoCount,*
UInt32 * *AttachedCertificatesCount,*
UInt32 * *CrlsCount);*

Назначение:

Получение информации об ЭП из CMS в файле.

Параметры:

ctxX	in	Контекст библиотеки
currentCmsFile	in	Файл CMS в формате ASN.1 или PEM
Data	out	Буфер с подписанными данными (данные присутствуют в CMS только в случае присоединенной ЭП)
DataLength	in/out	Вход - максимально возможный размер буфера с подписанными данными, выход - длина буфера с подписанными данными
Version	out	Версия CMS
AlgorithmsCount	out	Количество алгоритмов хеширования в CMS
SignerInfoCount	out	Количество ЭП в CMS
AttachedCertificatesCount	out	Количество присоединенных сертификатов в CMS
CrlsCount	out	Количество присоединенных стоп-листов в CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано

Описание функции:

Получает информации об ЭП с номером SignerNumber из CMS в файле.

7.1.18. *Получение времени ЭП и выделенного имени подписанта CMS в буфере*

Int32 DECL cr_GetCmsTimeCn(**H_CTX** *ctxX,*
UInt32 *SignatureNumber,*
UInt8 * *encodedCms,*
UInt32 *encodedCmsLength,*
H_TIMER *h_signingTime,*
char * *ansiString,*
UInt32 * *ansiStringLength);*

Назначение:

Получение времени ЭП и выделенного имени подписанта CMS в буфере.

Параметры:

ctxX	in	Контекст библиотеки
SignatureNumber	in	Номер ЭП в CMS (нумерация начинается с 1)
encodedCms	in	Буфер CMS
encodedCmsLength	in	Длина буфера CMS
h_signingTime	out	Время формирования ЭП с номером SignatureNumber
ansiString	out	Буфер с именем подписанта в кодировке Win-1251, с терминирующим 0
ansiStringLength	in/out	Вход - максимально возможный размер буфера с именем подписанта, выход - длина буфера с именем подписанта

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1

ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_INVALID_TIME	Неверное время

Описание функции:

Получает время создания ЭП и выделенное имя подписанта для ЭП с номером SignatureNumber для CMS в буфере.

7.1.19. *Получение времени ЭП и выделенного имени подписанта CMS в файле*

Int32 DECL cr_GetCmsFileTimeCn(**H_CTX** *ctxX,*
UInt32 *SignatureNumber,*
const char * *currentCmsFile,*
H_TIMER *h_signingTime,*
char * *ansiString,*
UInt32 * *ansiStringLength);*

Назначение:

Получение времени ЭП и выделенного имени подписанта CMS в файле.

Параметры:

ctxX	in	Контекст библиотеки
SignatureNumber	in	Номер ЭП в файле CMS
currentCmsFile	in	Файл CMS
h_signingTime	out	Время формирования ЭП с номером SignatureNumber
ansiString	out	Буфер с именем подписанта в кодировке Win-1251, с терминирующим 0
ansiStringLength	in/out	Вход - максимально возможный размер буфера с именем подписанта, выход - длина буфера с именем подписанта

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_INVALID_TIME	Неверное время

Описание функции:

Получает время создания ЭП и выделенное имя подписанта для ЭП с номером SignatureNumber для CMS в файле.

7.1.20. *Получение времени ЭП и выделенного имени создателя списка отозванных сертификатов*

Int32 DECL cr_GetCrlTimeCn(**H_CTX** *ctxX,*
UInt8 * *encodedCrl,*
UInt32 *encodedCrlLength,*
H_TIMER *timerThisUpdate,*

H_TIMER *timerNextUpdate,*
char * *ansiString,*
UInt32 * *ansiStringLength);*

Назначение:

Получение времени ЭП и выделенного имени создателя списка отозванных сертификатов.

Параметры:

ctxX	in	Контекст библиотеки
encodedCrl	in	Буфер стоп-листа в формате ASN.1 или PEM
encodedCrlLength	in	Длина буфера стоп-листа в формате ASN.1 или PEM
timerThisUpdate	out	Время формирования стоп-листа
timerNextUpdate	out	Время формирования следующего стоп-листа
ansiString	out	Буфер с именем подписанта в кодировке Win-1251, с терминирующим 0
ansiStringLength	in/out	Вход - максимально возможный размер буфера с именем подписанта, выход - длина буфера с именем подписанта

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта

ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время

Описание функции:

Получает время создания ЭП и выделенное имя создателя списка отозванных сертификатов.

7.1.21. *Получение времени ЭП и выделенного имени владельца из сертификата в буфере*

Int32 DECL cr_GetCertificateTimeCn(**H_CTX** *ctxX,*
UInt8 * *encodedCertificate,*
UInt32 *encodedCertificateLength,*
H_TIMER *validNotBefore,*
H_TIMER *validNotAfter,*
char * *ansiString,*
UInt32 * *ansiStringLength);*

Назначение:

Получение времени ЭП и выделенного имени владельца из сертификата в буфере.

Параметры:

ctxX	in	Контекст библиотеки
encodedCertificate	in	Буфер сертификата в формате ASN.1 или PEM
encodedCertificateLength	in	Длина буфера сертификата
validNotBefore	out	Время начала срока действия сертификата
validNotAfter	out	Время окончания срока действия сертификата
ansiString	out	Буфер с именем владельца в кодировке Win-1251, с терминирующим 0
ansiStringLength	in/out	Вход - максимально возможный размер буфера с именем владельца, выход - длина буфера с именем владельца

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время

Описание функции:

Получает время ЭП и выделенное имя владельца сертификата в буфере.

7.1.22. *Получение времени ЭП и выделенного имени владельца из сертификата в хранилище*

Int32 DECL cr_GetCertificateTimeCnByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN,*
H_TIMER *validNotBefore,*
H_TIMER *validNotAfter,*
char * *ansiString,*
UInt32 * *ansiStringLength);*

Назначение:

Получение времени ЭП и выделенного имени владельца из сертификата в хранилище.

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище
validNotBefore	out	Время начала срока действия сертификата
validNotAfter	out	Время окончания срока действия сертификата

ansiString	out	Буфер с именем владельца в кодировке Win-1251, с терминирующим 0
ansiStringLength	in/out	Вход - максимально возможный размер буфера с именем владельца, выход - длина буфера с именем владельца

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_INVALID_TIME	Неверное время
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_MEM	Недостаточно памяти
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана

Описание функции:

Получает время ЭП и выделенное имя владельца сертификата, содержащегося в хранилище.

7.1.23. *Завершение работы*

Int32 DECL cr_Finalize(**void)**

Назначение:

Завершение работы, освобождение ресурсов.

Параметры:

	none	Нет параметров

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

При завершении работы возвращает либо ERR_OK, либо количество незакрытых контекстов плюс 1000.

7.1.24. *Вызов конфигуратора*

```
void CALLBACK cfg(HWND hwnd,
                  HINSTANCE hInstance,
                  LPSTR lpszCmdLine,
                  int nCmdShow);
```

Описание функции:

Функция вызывает конфигуратор, см. «Конфигуратор «Бикрипт 5.0» Руководство пользователя 11485466.5014.044 34».
Функция доступна только в среде ОС семейства Windows.

7.2. Функции для реализации электронной подписи в соответствии со стандартом X.509

7.2.1. *Создание ключевой пары на основе выделенного имени и атрибутов, кодированных в формате ASN.1*

```
Int32 DECL cr_GenerateKeyPair(H_CTX ctxX,
                              UInt8* distinguished_name,
                              UInt32 distinguished_nameLength,
                              UInt8* attributes,
                              UInt32 attributesLength,
                              const char *userid,
                              char param_key,
                              char param_hash,
                              CARRIER_TYPE carrier,
                              UInt8 *master_key,
                              int master_len,
                              char* password,
                              UInt32 *passwordLength,
                              UInt8 *pkcs10_binary,
                              UInt32 *pkcs10Length);
```

Назначение:

Создание ключевой пары и запроса на сертификат для УЦ на основе выделенного имени и атрибутов, кодированных в формате ASN.1.

Параметры:

ctxX	in	Контекст библиотеки
distinguished_name	in	Буфер содержащий выделенное имя в формате ASN.1
distinguished_nameLength	in	Длина буфера выделенного имени
attributes	in	Буфер содержащий атрибуты в формате ASN.1
attributesLength	in	Длина буфера атрибутов
userid	in	Идентификатор ключа БиКрипт (не более 32 символов)
param_key	in	Параметры эллиптической кривой (см. Приложение 3)
param_hash	in	Параметры хеш-функции (см. Приложение 3)
carrier	in	Носитель для записи закрытого ключа (см. Приложение 3)
master_key	in	Равно NULL в случае ключа с паролем, буфер с мастер-ключом - в случае двухкомпонентного ключа, при этом главный ключ должен располагаться на том же носителе
master_len	in	Равно 0 в случае ключа с паролем, длина буфера с мастер-ключом (обычно 108) в случае двухкомпонентного ключа
password	in/out	Пароль для секретного ключа в случае ключа с паролем (6 символов и терминирующий 0) или пароль главного ключа ("" для пустого пароля)
passwordLength	in/out	Вход - максимально возможный размер поля password, выход - длина пароля в случае ключа с паролем, поле игнорируется в случае двухкомпонентного ключа
pkcs10_binary	out	Буфер запроса на сертификат, в формате PEM
pkcs10Length	in/out	Вход - максимально возможный размер буфера запроса на сертификат, выход - длина буфера сгенерированного запроса на сертификат

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_MEM	Недостаточно памяти

ERR_OPEN_FILE	Ошибка открытия файла
ERR_TOO_MANY	Слишком много носителей в USB устройстве, поддерживается работа только с одним
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_WRITE_FILE	Ошибка записи файла
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа

ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_BAD_SIGN	ЭП не верна

Описание функции:

Функция создает ключевую пару и запрос на сертификат для УЦ. Генерация секретного ключа происходит из ПДСЧ, который должен быть загружен путем вызова cr_LoadPrnd

7.2.2. ***Создание ключевой пары на основе данных, идентифицирующих владельца сертификата***

Int32 DECL cr_GenerateKeyPairByQualNames(**int** *A_flag_QUAL*,
**const char ***cert_req_Strana,*
**const char ***cert_req_Locality,*
**const char ***cert_req_Address,*
**const char ***cert_req_Region,*
**const char ***cert_req_OGRN,*
**const char ***cert_req_INN,*
**const char ***cert_req_Podrazdelenie,*
**const char ***cert_req_SNILS,*
**const char ***cert_req_SerialNomer,*
**const char ***cert_req_Family,*
**const char ***cert_req_Name,*
**const char ***cert_req_Dolzhnost,*
**const char ***cert_req_Organizat,*
**const char ***cert_req_Email,*
**const char ***Auserid,*
char *param_key*,
char *param_hash*,
CARRIER_TYPE *carrier*,

UInt8 * *master_key,*
int *master_len,*
char* *password,*
UInt32 * *passwordLength,*
UInt8 * *PKCS10Buffer,*
UInt32 * *PKCS10BufferLength);*

Назначение:

Создание ключевой пары и запроса на сертификат для УЦ на основе данных, идентифицирующих владельца сертификата.

Параметры:

A_flag_QUAL	in	0 - неквалифицированный запрос на сертификат, 1 - квалифицированный
cert_req_Strana	in	Страна, в кодировке Win-1251
cert_req_Locality	in	Город, в кодировке Win-1251
cert_req_Address	in	Адрес, в кодировке Win-1251
cert_req_Region	in	Регион, в кодировке Win-1251
cert_req_OGRN	in	ОГРН, в кодировке Win-1251
cert_req_INN	in	ИНН, в кодировке Win-1251
cert_req_Podrazdelenie	in	Наименование подразделения (отдела), в кодировке Win-1251
cert_req_SNILS	in	СНИЛС, в кодировке Win-1251
cert_req_SerialNomer	in	Серийный номер (носителя), в кодировке Win-1251
cert_req_Family	in	Фамилия, в кодировке Win-1251
cert_req_Name	in	Имя отчество, в кодировке Win-1251
cert_req_Dolzhnost	in	Должность, в кодировке Win-1251
cert_req_Organizat	in	Организация, в кодировке Win-1251
cert_req_Email	in	e-mail, в кодировке Win-1251
Auserid	in	Идентификатор ключа БиКрипт (не более 32 байт), в кодировке Win-1251
param_key	in	Параметры эллиптической кривой (см. Приложение 3)
param_hash	in	Параметры хеш-функции (см. Приложение 3)
carrier	in	Носитель для записи закрытого ключа (см. Приложение 3)
master_key	in	Равно NULL в случае ключа с паролем, буфер с мастер-ключом - в случае двухкомпонентного ключа, при этом главный ключ должен располагаться на том же носителе
master_len	in	Равно 0 в случае ключа с паролем, длина буфера с мастер-ключом (обычно 108) в случае двухкомпонентного ключа
password	in/out	Пароль для секретного ключа в случае ключа с паролем (6 символов и терминирующий 0) или пароль главного ключа ("" для пустого пароля)

passwordLength	in/out	Вход - максимально возможный размер поля password, выход - длина пароля в случае ключа с паролем, поле игнорируется в случае двухкомпонентного ключа
PKCS10Buffer	out	Буфер запроса на сертификат в формате PEM
PKCS10BufferLength	in/out	Вход - максимально возможный размер запроса на сертификат, выход - длина сгенерированного запроса на сертификат

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_OPEN_FILE	Ошибка открытия файла
ERR_TOO_MANY	Слишком много носителей в USB устройстве, поддерживается работа только с одним
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа

ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_WRITE_FILE	Ошибка записи файла
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_READ_FILE	Ошибка чтения файла
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_BAD_SIGN	ЭП не верна

Описание функции:

Функция создает ключевую пару и запрос на сертификат для УЦ. Генерация секретного ключа происходит из ПДСЧ, который должен быть загружен путем вызова cr_LoadPrnd#

7.2.3. ***Инициализация контекста библиотеки***

Int32 DECL cr_CreateContext(**H_TIMER** *timer,*
H_CTX* *ctx);*

Назначение:

Создание контекста работы библиотеки

Параметры:

timer	in	Заполненный контекст таймера или NULL для текущего момента времени
ctx	out	Созданный контекст библиотеки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM

Описание функции:

Инициализация внутренних ресурсов библиотеки. Опциональная переменная timer указывает время, в которое создается контекст. Если переменная не указана (NULL), то в ходе работы библиотеки используется текущее системное время. Соответственно, указанное, либо текущее время используется в процессе загрузки сертификатов, СОС, проверки ЭП и т.д.

7.2.4. *Добавление сертификата в контекст библиотеки*

Int32 DECL cr_AddCertificate(**H_CTX** *ctx,*
const char * *Filename,*
UInt8* *binaryCert,*
UInt32 *binaryCertLength,*
H_STORAGE * *itemN);*

Назначение:

Функция позволяет добавить сертификат в контекст библиотеки.

Параметры:

ctx	in	Контекст библиотеки
Filename	in	Имя файла сертификата, если сертификат в файле или NULL если сертификат в буфере
binaryCert	in	Буфер сертификата или NULL если сертификат в файле
binaryCertLength	in	Длина буфера сертификата или 0 если сертификат в файле
itemN	out	Индекс в хранилище сертификатов

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана

ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные

Описание функции:

Этой функцией допускается загружать корневые сертификаты, УЦ сертификаты (с выставленным в атрибуте KeyUsage признаком CA), а также пользовательские сертификаты.

7.2.5. *Добавление списка отозванных сертификатов в контекст библиотеки*

Int32 DECL cr_AddCrl(**H_CTX** *ctx,*
const char * *Filename,*
UInt8* *binary_crl,*
UInt32 *crlLength,*
H_STORAGE * *itemN);*

Назначение:

Функция позволяет добавить СОС в контекст библиотеки.

Параметры:

ctx	in	Контекст библиотеки
Filename	in	Имя файла стоп-листа
binary_crl	in	Буфер стоп-листа или NULL если стоп-лист в файле
crlLength	in	Длина буфера стоп-листа или 0 если стоп-лист в файле
itemN	out	Индекс сертификата подписанта в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна

ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе

Описание функции:

При добавлении СОС происходит проверка его ЭП по внутреннему хранилищу контекста библиотеки, сроков действия, наличия более новых СОС того же издателя. Если хотя бы одно из условий не выполняется, метод возвращает ошибку загрузки.

7.2.6. ***Добавление сертификатов и списков отозванных сертификатов из файлов в контекст библиотеки***

Int32 DECL cr_AddCertificateAndCrlByPath(**H_CTX** *ctx,*
const char * *certPathMask,*
const char * *crlPathMask,*
UInt32 * *numof_ok_certfiles,*
UInt32 * *numof_err_certfiles,*
UInt32 * *numof_ok_crlfiles,*
UInt32 * *numof_err_crlfiles);*

Назначение:

Добавление сертификатов и списков отозванных сертификатов из файлов в директории по маске в контекст библиотеки.

Параметры:

ctx	in	Контекст библиотеки
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
numof_ok_certfiles	out	Количество успешно добавленных сертификатов (в том числе не проверенных)
numof_err_certfiles	out	Количество сертификатов, удовлетворяющих маске, но по каким-либо причинам не добавленных в хранилище
numof_ok_crlfiles	out	Количество успешно добавленных сертификатов, все стоп-листы добавляются после полной проверки
numof_err_crlfiles	out	Количество стоп-листов, удовлетворяющих маске, но по каким-либо причинам не добавленных в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе

Описание функции:

Добавляет сертификаты и стоп-листы из директории. При этом в соответствующих полях можно получить количество успешно обработанных файлов и файлов с ошибкой. Может загружать сертификаты из файлов P7B, при этом возвращается

7.2.7. *Инициализация ПДСЧ*

Int32 DECL cr_LoadPrnd(**H_CTX** *ctx,*
int *flag_init_grn);*

Назначение:

Инициализация ПДСЧ

Параметры:

ctx	in	Контекст библиотеки
flag_init_grn	in	В случае ошибки при чтении файла с ключом ПДСЧ: если flag_init_grn = 1, то для инициализации ПКДСЧ будет автоматически использован ПДСЧ из библиотеки grn.dll, grn64.dll, libbiogrn.dylib или libbiogrn_64.dylib если flag_init_grn = 0, то библиотеки grn.dll, grn64.dll, libbiogrn.dylib и libbiogrn_64.dylib не будут использоваться и будет выдана ошибка

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_DSCH	Ошибка ПДСЧ
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_BAD_CRYP	Ошибка шифрования
ERR_READ_FILE	Ошибка чтения файла
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Загрузка ключа ПДСЧ с носителя. В случае flag_init_grn=1 и в случае отсутствия носителя или ключа на носителе будет показана экранная форма для создания ключа ПДСЧ с помощью движений мыши.

7.2.8. ***Создание копии контекста библиотеки***

Int32 DECL cr_DupContext(**H_CTX** *ctx,*
H_CTX* *ctx_out);*

Назначение:

Создание копии контекста библиотеки

Параметры:

ctx	in	Контекст библиотеки
ctx_out	out	Созданный контекст библиотеки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта

ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1

Описание функции:

Создание новой копии контекста библиотеки по существующей. При этом все сертификаты и стоп-листы также будут скопированы в новый контекст.

7.2.9. *Освобождение контекста библиотеки*

Int32 DECL cr_FreeContext(**H_CTX** *ctx);*

Назначение:

Освобождение контекста библиотеки.

Параметры:

ctx	in	Контекст библиотеки

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Завершение работы с контекстом библиотеки. При этом области памяти, занятые секретными ключами и ключом ПДСЧ, использованными в контексте, обнуляются.

7.2.10. *Получение количества ключевых пар в контексте библиотеки*

Int32 DECL cr_GetNumOfKeyPair(**H_CTX** *ctx,*
UInt32 * *numof_keypair);*

Назначение:

Получение количества ключевых пар в контексте библиотеки.

Параметры:

ctx	in	Контекст библиотеки
numof_keypair	out	Количество ключевых пар в конфигурационном файле

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Получает количество ключевых пар в конфигурационном файле.

7.2.11. *Получение информации о ключевой паре в контексте библиотеки*

Int32 DECL cr_GetKeyPairInfo(**H_CTX** *ctx,*
UInt32 *keypairNumber,*
char* *keypairLabel,*
UInt32 * *keypairLabelLength,*
CARRIER_TYPE* *carrier,*
char * *filenameCertificate,*
UInt32 * *filenameCertificateLength,*
UInt8* *binaryCert,*
UInt32 * *binaryCertLength,*
UInt32 * *certificateInChainToAdd);*

Назначение:

Получение информации о ключевой паре в контексте библиотеки.

Параметры:

ctx	in	Контекст библиотеки
keypairNumber	in	Номер ключевой пары, (первый номер 0)
keypairLabel	out	Буфер с идентификатором ключевой пары
keypairLabelLength	in/out	Вход - максимально возможный размер буфера с идентификатором ключевой пары, выход - длина буфера с идентификатором ключевой пары
carrier	out	Носитель закрытого ключа (см. приложение №3)
filenameCertificate	out	Имя файла сертификата

filenameCertificateLength	in/out	Вход - максимально возможный размер буфера файла сертификата, выход - длина буфера файла сертификата
binaryCert	out	Буфер с сертификатом в формате ASN.1
binaryCertLength	in/out	Вход - максимально возможный размер буфера с сертификатом, выход - длина буфера с сертификатом
certificateInChainToAdd	out	Количество сертификатов присоединяемых в CMS при формировании ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Необходимо вызывать перед показом пользователю сообщения о необходимости приложить ТМ-идентификатор или вставить флешку с ключом. Идентификатор ключевой пары должен содержаться в сообщении. После показа сообщения следует вызвать функцию cr_LoadKeyPair.

7.2.12. *Получение информации о носителе ключа ПДСЧ в контексте библиотеки*

Int32 DECL cr_GetPrndCarrier(**H_CTX** *ctx,*
CARRIER_TYPE* *carrier);*

Назначение:

Получение информации о носителе ключа ПДСЧ в контексте библиотеки

Параметры:

ctx	in	Контекст библиотеки
carrier	out	Носитель ключа ПДСЧ (см. приложение №3)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_FOUND	Поиск не оказался результативным

Описание функции:

Необходимо вызывать перед показом пользователю сообщения о необходимости приложить ТМ-идентификатор или вставить флешку с ключом. После показа сообщения следует вызвать функцию cr_LoadPrnd.

7.2.13. *Поиск ключевой пары по сертификату в контексте библиотеки*

Int32 DECL cr_FindKeyPair (**H_CTX** *ctx,*
UInt8* *binaryCert,*
UInt32 *binaryCertLength,*
UInt32* *keypairNumber);*

Назначение:

Поиск ключевой пары по сертификату в контексте библиотеки.

Параметры:

ctx	in	Контекст библиотеки
binaryCert	in	Буфер сертификата в формате ASN.1 или PEM
binaryCertLength	in	Длина буфера сертификата
keypairNumber	out	Номер связки (нумерация начинается с 0)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Функция ищет в конфигурационном файле и возвращает номер связки, содержащий заданный сертификат.

7.2.14. *Загрузка ключевой пары в контексте библиотеки*

Int32 DECL cr_LoadKeyPair(**H_CTX** *ctx,*
UInt32 *keypairNumber,*
**const char ***Apassw,*
H_KEYPAIR* *keypair);*

Назначение:

Загружает секретный ключ с носителя, и сертификат секретного ключа из конфигурационного файла.

Параметры:

ctx	in	Контекст библиотеки
keypairNumber	in	Номер связки (нумерация начинается с 0)
Apassw	in	В случае ключа с паролем передается пароль ключа, в случае двухкомпонентного ключа передается пароль главного ключа.
keypair	out	Контекст ключевой пары

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_TMDRV_NOT_FOUND	Не найден драйвер TMDRV
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_GKUZ_PSW	Ошибка пароля главного ключа
ERR_NO_TM_ATTACHED	ТМ-идентификатор не приложен к съемнику
ERR_READ_TM	Ошибка чтения ТМ-идентификатора
ERR_GK_READ	Ошибка загрузки главного ключа
ERR_UZ_READ	Ошибка чтения узлов замены
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_CRC_SK_FILE	Ошибка контрольной суммы закрытого ключа
ERR_NOT_USED_TMDRV	Требуется драйвер TMDRV, но он не был проинициализирован ранее
ERR_NOT_USED_GKUZ	Требуется главный ключ, но он не был проинициализирован ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_CRYDRV	Внутренняя ошибка работы криптобиблиотеки, работа невозможна
ERR_READ_MK_FILE	Ошибка чтения файла с маской
ERR_INIT_CRYMASK	Ошибка контрольной суммы файла с маской
ERR_WRITE_TM	Ошибка записи ТМ-идентификатора
ERR_CRC_TM	Ошибка контрольной суммы ТМ-идентификатора
ERR_NOT_SUPPORTED	Функция не поддерживается в данной версии или в данном исполнении
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1

ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные

Описание функции:

Информация о носителе секретного ключа и главного ключа берется из конфигурационного файла. Нумерация номеров связки начинается с нуля. После использования ключевой пары следует освободить контекст путем вызова функции cr_FreeKeyPair.

7.2.15. *Освобождение ресурсов ключевой пары*

Int32 DECL cr_FreeKeyPair(**H_KEYPAIR** *keypair);*

Назначение:

Освобождает контекст ключевой пары.

Параметры:

keypair	in	Контекст ключевой пары, возвращенный функцией cr_LoadKeyPair

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Освобождает ресурсы контекста ключевой пары после использования, с очисткой всех областей ОЗУ, в которых содержались криптографическая информация. Если не освободить контекст ключевой пары,

функция cr_Finalize вернет код ошибки 1001. Если не освободить два контекста ключевой пары, функция cr_Finalize вернет код ошибки 1002 и т.д.

7.2.16. *Инициализация контекста вычисления значений хеш-функции*

Int32 DECL cr_DigestInit(**H_CTX** *ctxX,*
H_DIGEST* *digest);*

Назначение:

Инициализация контекста вычисления значений хеш-функции.

Параметры:

ctxX	in	Контекст библиотеки
digest	out	Контекст хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Начало работы с вычислением значения хеш-функции.

7.2.17. *Задание параметров контекста вычисления значений хеш-функции*

Int32 DECL cr_SetDigestParam(**H_DIGEST** *digest,*
char *param);*

Назначение:

Задание параметров контекста вычисления значений хеш-функции.

Параметры:

digest	in	Контекст хеш-функции
param	in	Параметры хеш-функции, см. приложение №3

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Значения параметров хеш-функции см. Приложение 3.

7.2.18. *Получение параметров контекста вычисления значений хеш-функции*

Int32 DECL cr_GetDigestParam(**H_DIGEST** *digest,*
int * *param);*

Назначение:

Получение параметров контекста вычисления значений хеш-функции.

Параметры:

digest	in	Контекст хеш-функции
param	out	Параметры хеш-функции, см. Приложение №3

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее

Описание функции:

Значения параметров хеш-функции см. Приложение 3.

7.2.19. *Задание параметров контекста вычисления значений хеш-функции, согласованных с ключевой парой*

Int32 DECL cr_SetDigestParamByKeyPair(**H_DIGEST** *digest,*
H_KEYPAIR *keypair);*

Назначение:

Задание параметров контекста вычисления значений хеш-функции, согласованных с ключевой парой

Параметры:

digest	in	Контекст хеш-функции
keypair	in	Контекст ключевой пары

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа

Описание функции:

Для формирования значения хеш-функции с нужными параметрами следует или установить эти параметры посредством cr_SetDigestParam или использовать параметры, хранящиеся в ключевой паре, которые в свою очередь были получены из сертификата открытого ключа. Значения параметров хеш-функции см. Приложение 3.

7.2.20. *Добавление порции данных в контекст вычисления значений хеш-функции*

Int32 DECL cr_DigestUpdate(**H_DIGEST** *digest,*
UInt8* *data_next,*
UInt32 *data_length);*

Назначение:

Добавление порции данных в контекст вычисления значений хеш-функции.

Параметры:

digest	in	Контекст хеш-функции
data_next	in	Буфер с данными
data_length	in	Длина буфера с данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя

Описание функции:

Удобно для потокового режима работы вычисления значений хеш-функции.

7.2.21. *Получение значения хеш-функции для буфера памяти*

Int32 DECL cr_DigestBuffer(**int** *Aparam,*
UInt8 **hashBuffer,*
UInt32 *hashBufferLength,*
UInt8 **hashResult,*
UInt32 **hashResultLength);*

Назначение:

Вычисляет значение хеш-функции для буфера памяти без контекста.

Параметры:

Aparam	in	Параметр хеша, *см. Приложение № 3*
hashBuffer	in	Буфер с данными, для которых вычисляется хеш-функция
hashBufferLength	In	Длина буфера с данными
hashResult	out	Результат хеш-функции
hashResultLength	in/out	На входе максимальная длина, на выходе результирующая длина буфера

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
Значение, отличное от нуля	Согласно Списку ошибок, *см. Приложение №1*

Описание функции:

Удобно для вычисления значений хеш-функции с помощью одной функции.

7.2.22. *Получение значения в контексте вычисления значений хеш-функции*

Int32 DECL cr_DigestFinal(**H_DIGEST** *digest,*
UInt8* *hashData,*
UInt32 * *hashDataLength);*

Назначение:

Получение значения в контексте вычисления значений хеш-функции.

Параметры:

digest	in	Контекст хеш-функции
hashData	out	Буфер с результирующим значением хеш-функции
hashDataLength	in/out	Вход - максимально возможный размер буфера с результирующим значением, выход - длина буфера с результирующим значением

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Полученное значение можно использовать для печати на экран значения хеш-функции.

7.2.23. *Закрыть контекст вычисления значений хеш-функции*

Int32 DECL cr_DigestClose(**H_DIGEST** *digest);*

Назначение:

Закрыть контекст вычисления значений хеш-функции.

Параметры:

digest	in	Контекст хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст

Описание функции:

Завершение работы с контекстом вычисления значений хеш-функции.

7.2.24. *Инициализация контекста операции формирования ЭП*

Int32 DECL cr_SignInit(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
UInt8* *unsigned_attrs,*
UInt32 *unsigned_attrs_length,*
UInt8* *signed_attrs,*
UInt32 *signed_attrs_length,*
SIGN_OP_TYPE *operation_type,*
H_CTX_SIGN* *s_ctx);*

Назначение:

Инициализация контекста операции формирования ЭП.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
unsigned_attrs	in	Неподписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
unsigned_attrs_length	in	Длина неподписанных атрибутов в формате ASN.1
signed_attrs	in	Подписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
signed_attrs_length	in	Длина подписанных атрибутов в формате ASN.1

operation_type	in	Перечисление типов создаваемого CMS: Op_CreateAttached (присоединенная), Op_CreateDetached (отсоединенная), Op_AddSign (добавить ЭП в существующую CMS)
s_ctx	out	Контекст операции формирования ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти

Описание функции:

При формировании ЭП происходит контроль времени действия сертификатов и списков отозванных сертификатов, участвующих в данной процедуре (сертификаты из цепочки подписанта и СОС, если таковые были загружены в контекст). Все они должны быть действительны на текущий момент времени. Иначе функция cr_SignInit вернет соответствующую ошибку. При этом если ни один СОС не был загружен в контекст, то это не является ошибкой. В таком случае будет осуществляться контроль действия только сертификатов из цепочки подписанта.

7.2.25. ***Добавление значения хеш-функции в контекст операции формирования ЭП***

Int32 DECL cr_SignPutHash(**H_CTX_SIGN** *s_ctx,*
H_DIGEST *digest);*

Назначение:

Добавление значения хеш-функции, вычисленной ранее.

Параметры:

s_ctx	in	Контекст операции формирования ЭП
digest	in	Контекст хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее

ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Добавляет значение хеш-функции к контексту операции формирования ЭП.

7.2.26. *Накопление данных в контексте операции формирования ЭП*

Int32 DECL cr_SignPutData(**H_CTX_SIGN** *s_ctx,*
UInt8* *data_part,*
UInt32 dataLength *);*

Назначение:

Накопление данных в контексте ЭП. При вызове cr_Sign, считаем, что данные закончены.

Параметры:

s_ctx	in	Контекст операции формирования ЭП
data_part	in	Буфер с подписываемыми данными
	in	Длина буфера с подписываемыми данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Накапливает подписываемые данные в контексте операции формирования ЭП.

7.2.27. *Добавление в контекст операции формирования ЭП существующей CMS*

Int32 DECL cr_SignPutCms(**H_CTX_SIGN** *s_ctx,*
UInt8* *cms_part,*
UInt32 *cmsLength);*

Назначение:

Добавить в контекст операции формирования ЭП CMS с определенной длиной (единым целым блоком). Метод используется для добавления ЭП к существующему CMS.

Параметры:

s_ctx	in	Контекст операции формирования ЭП
cms_part	in	Буфер CMS
cmsLength	in	Длина буфера CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Добавляет CMS в контекст операции формирования ЭП.

7.2.28. *Формирование ЭП в контексте операции формирования ЭП*

Int32 DECL cr_Sign(**H_CTX_SIGN** *s_ctx,*
UInt8* *cmsSignedData,*
UInt32 *cmsSignedDataLength *);*

Назначение:

Подписать блок данных. Также добавить ЭП к существующему CMS, предварительно загруженному в контекст операции формирования ЭП.

Параметры:

s_ctx	in	Контекст операции формирования ЭП
cmsSignedData	out	Буфер с результирующей CMS
	in/out	Вход - максимально возможный размер буфера с результирующей CMS, выход - длина буфера с результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_CERTIFICATES_COUNT	Построена слишком короткая цепочка, число сертификатов не соответствует реальной длине
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_INVALID_TIME	Неверное время
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_DIFFERENT	Сравниваемые объекты различны

ERR_ALREADY	Уже присутствует

Описание функции:

Функция создает результирующий CMS на основе добавленных ранее данных или значения хеш-функции. А также ранее созданного CMS, если это вторая или последующие ЭП.

7.2.29. *Закрытие контекста операции формирования ЭП*

Int32 DECL cr_SignClose(**H_CTX_SIGN** *s_ctx);*

Назначение:

Завершение контекста операции формирования ЭП

Параметры:

s_ctx	in	Контекст операции формирования ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Завершение работы с контекстом операции формирования ЭП. Все ресурсы, задействованные в контексте, освобождаются.

7.2.30. *Электронная подпись области памяти*

Int32 DECL cr_SignBuffer(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
UInt8* *unsigned_attrs,*
UInt32 *unsigned_attrs_length,*
UInt8* *signed_attrs,*
UInt32 *signed_attrs_length,*
int *flag_ATTACHED_CMS,*
UInt8 * *dataBuffer,*
UInt32 *dataBufferLength,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt8 * *signResultPkcs7,*
UInt32 * *signResultPkcs7Length);*

Назначение:

Электронная подпись буфера.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
unsigned_attrs	in	Неподписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
unsigned_attrs_length	in	Длина неподписанных атрибутов в формате ASN.1 или 0 если дополнительных атрибутов нет
signed_attrs	in	Подписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
signed_attrs_length	in	Длина подписанных атрибутов в формате ASN.1 или 0 если дополнительных атрибутов нет
flag_ATTACHED_CMS	in	0 - отсоединенная ЭП, 1 - присоединенная ЭП
dataBuffer	in	Буфер с подписываемыми данными
dataBufferLength	in	Длина буфера с подписываемыми данными
encodedCurrentCms	in	Если первая ЭП, тогда NULL. Если создается вторая и последующие ЭП, тогда буфер CMS с предыдущими ЭП
encodedCurrentCmsLength	in	Если первая ЭП, тогда ноль. Если создается вторая и последующие ЭП, тогда длина буфера CMS с предыдущими ЭП
signResultPkcs7	out	Буфер с результирующей CMS
signResultPkcs7Length	in/out	Вход - максимально возможный размер буфера с результирующей CMS, выход - длина буфера с результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер

ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_CERTIFICATES_COUNT	Построена слишком короткая цепочка, число сертификатов не соответствует реальной длине
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_INVALID_TIME	Неверное время
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_ALREADY	Уже присутствует

Описание функции:

Создает ЭП данных из буфера dataBuffer, ЭП и сертификаты помещаются в буфер с результирующей CMS. Если flag_ATTACHED_CMS=1, в буфер с результирующей CMS также помещаются подписываемые данные. Количество присоединяемых сертификатов задается в конфигурационном файле в разделе связок.

7.2.31. *Электронная подпись файла*

Int32 DECL cr_SignFile(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
UInt8* *unsigned_attrs,*
UInt32 *unsigned_attrs_length,*

UInt8* *signed_attrs,*
UInt32 *signed_attrs_length,*
int *flag_ATTACHED_CMS,*
**const char *****dataFilename,*
**const char *****currentCmsFile,*
**const char *****resultCmsFile);*

Назначение:

Электронная подпись файла.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
unsigned_attrs	in	Неподписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
unsigned_attrs_length	in	Длина неподписанных атрибутов в формате ASN.1 или 0 если дополнительных атрибутов нет
signed_attrs	in	Подписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
signed_attrs_length	in	Длина подписанных атрибутов в формате ASN.1 или 0 если дополнительных атрибутов нет
flag_ATTACHED_CMS	in	0 - отсоединенная ЭП, 1- присоединенная ЭП
dataFilename	in	Имя файла с подписываемыми данными, в случае отсоединенной ЭП или если первая присоединенная ЭП, NULL в противном случае
currentCmsFile	in	Имя файла CMS с предыдущими ЭП или NULL если это первая ЭП
resultCmsFile	in	Имя файла с результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_MEM	Недостаточно памяти
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_NOT_FOUND	Поиск не оказался результативным

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_CERTIFICATES_COUNT	Построена слишком короткая цепочка, число сертификатов не соответствует реальной длине
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_INVALID_TIME	Неверное время
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_ALREADY	Уже присутствует
ERR_WRITE_FILE	Ошибка записи файла

Описание функции:

Создает ЭП данных из файла dataFilename, ЭП и сертификаты помещаются в файл с результирующей CMS. Если flag_ATTACHED_CMS=1 в файл с результирующей CMS также помещаются подписываемые данные. Количество присоединяемых сертификатов задается в конфигурационном файле в разделе связок.

7.2.32. *Электронная подпись значения хеш-функции*

Int32 DECL cr_SignHash(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
UInt8* *unsigned_attrs,*
UInt32 *unsigned_attrs_length,*
UInt8* *signed_attrs,*
UInt32 *signed_attrs_length,*
H_DIGEST *hash,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt8 * *signResultPkcs7,*
UInt32 * *signResultPkcs7Length);*

Назначение:

Подпись значения хеш-функции, содержащегося в контексте, созданного функцией cr_HashOpen. При этом ЭП всегда отсоединенная. Для создания присоединенной ЭП следует использовать функцию cr_SignBuffer.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
unsigned_attrs	in	Неподписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
unsigned_attrs_length	in	Длина неподписанных атрибутов в формате ASN.1
signed_attrs	in	Подписанные атрибуты в формате ASN.1 или NULL если дополнительных атрибутов нет
signed_attrs_length	in	Длина подписанных атрибутов в формате ASN.1
hash	in	Контекст хеш-функции
encodedCurrentCms	in	Если первая ЭП, тогда NULL. Если создается вторая и последующие ЭП, тогда буфер CMS с предыдущими ЭП
encodedCurrentCmsLength	in	Если первая ЭП, тогда ноль. Если создается вторая и последующие ЭП, тогда длина буфера CMS с предыдущими ЭП
signResultPkcs7	out	Буфер с результирующей CMS
signResultPkcs7Length	in/out	Вход - максимально возможный размер буфера с результирующей CMS, выход - длина буфера с результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_CERTIFICATES_COUNT	Построена слишком короткая цепочка, число сертификатов не соответствует реальной длине
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_INVALID_TIME	Неверное время
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_DIFFERENT	Сравниваемые объекты различны

ERR_ALREADY	Уже присутствует

Описание функции:

Создает CMS с ЭП, используя значение хеш-функции, заранее рассчитанное функциями СКЗИ, ЭП и сертификаты помещаются в буфер с результирующей CMS. Количество присоединяемых сертификатов задается в конфигурационном файле в разделе связок.

7.2.33. *Инициализация контекста операции проверки ЭП*

Int32 DECL cr_CheckInit(**H_CTX** *ctx,*
H_CTX_CHECK* *c_ctx);*

Назначение:

Инициализировать контекст операции проверки ЭП.

Параметры:

ctx	in	Контекст библиотеки
c_ctx	out	Контекст операции проверки ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти

Описание функции:

Начало работы с контекстом операции проверки ЭП.

7.2.34. *Добавление значения хеш-функции в контекст операции проверки ЭП*

Int32 DECL cr_CheckPutHash(**H_CTX_CHECK** *c_ctx,*
H_DIGEST *digest);*

Назначение:

Добавление значения хеш-функции в контекст операции проверки ЭП.

Параметры:

c_ctx	in	Контекст операции проверки ЭП
digest	in	Контекст хеш-функции

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Данную функцию необходимо использовать только в случае отсоединенной ЭП.

7.2.35. *Накопление данных для проверки отсоединенной ЭП в контексте операции проверки ЭП*

Int32 DECL cr_CheckPutData(**H_CTX_CHECK** *c_ctx,*
UInt8* *data_part,*
UInt32 *dataLength);*

Назначение:

Накопление данных для проверки ЭП (поддержка потокового режима).

Параметры:

c_ctx	in	Контекст операции проверки ЭП
data_part	in	Буфер с проверяемыми данными, если это отсоединенная ЭП
dataLength	in	Длина буфера с проверяемыми данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Данную функцию необходимо использовать только в случае отсоединенной ЭП.

7.2.36. *Добавление в контекст операции проверки ЭП существующей CMS*

Int32 DECL cr_CheckPutCms(**H_CTX_CHECK** *c_ctx,*
UInt8* *cms_part,*
UInt32 *cmsLength);*

Назначение:

Добавление в контекст операции проверки ЭП существующей CMS (поддержка потокового режима).

Параметры:

c_ctx	in	Контекст операции проверки ЭП
cms_part	in	Буфер CMS с проверяемыми ЭП
cmsLength	in	Длина буфера CMS с проверяемыми ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Данную функцию необходимо использовать только в случае второй и последующих ЭП.

7.2.37. *Проверка ЭП и получение результатов в контексте операции проверки ЭП*

Int32 DECL cr_Check(**H_CTX_CHECK** *c_ctx,*
**UInt32 ***resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN]);*

Назначение:

Проверка ЭП для буфера CMS.

Параметры:

c_ctx	in	Контекст операции проверки ЭП

resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_MEM	Недостаточно памяти
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа

ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш
ERR_INVALID_TIME	Неверное время
ERR_OPEN_FILE	Ошибка открытия файла
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе

Описание функции:

Проверка электронной подписи CMS для присоединенной или отсоединенной ЭП. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.38. *Закрытие контекста операции проверки ЭП*

Int32 DECL cr_CheckClose(**H_CTX_CHECK** *c_ctx);*

Назначение:

Закрытие контекста операции проверки ЭП.

Параметры:

c_ctx	in	Контекст операции проверки ЭП

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Завершение работы с контекстом операции проверки ЭП.

7.2.39. *Проверка ЭП области памяти*

Int32 DECL cr_CheckBuffer(**H_TIMER** *timer,*
const char * *certPathMask,*
const char * *crlPathMask,*
UInt8 * *dataBuffer,*
UInt32 *dataBufferLength,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt32 * *resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN]);*

Назначение:

Проверка ЭП области памяти.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
dataBuffer	in	Буфер с проверяемыми данными, в случае отсоединённой ЭП или NULL в случае присоединенной ЭП
dataBufferLength	in	Длина буфера с проверяемыми данными
encodedCurrentCms	in	Буфер CMS с проверяемыми ЭП

encodedCurrentCmsLength	in	Длина буфера CMS с проверяемыми ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер

ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи буфера CMS для присоединенной или отсоединенной ЭП. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.40. *Проверка ЭП области памяти*

Int32 DECL cr_CheckBuffer2(**H_TIMER** *timer,*
const char * *certPathMask,*
const char * *crlPathMask,*
UInt8 * *dataBuffer,*
UInt32 *dataBufferLength,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt32 * *resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN],*
H_CTX **h_ctx);*

Назначение:

Проверка ЭП области памяти.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
dataBuffer	in	Буфер с проверяемыми данными, в случае отсоединённой ЭП или NULL в случае присоединенной ЭП
dataBufferLength	in	Длина буфера с проверяемыми данными
encodedCurrentCms	in	Буфер CMS с проверяемыми ЭП
encodedCurrentCmsLength	in	Длина буфера CMS с проверяемыми ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
h_ctx	out	Контекст хранения результирующих сертификатов (требует последующего удаления путем cr_FreeContext)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем

ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи буфера CMS для присоединенной или отсоединенной ЭП. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.41. *Проверка ЭП файла*

Int32 DECL cr_CheckFile(**H_TIMER** *timer,*
const char * *certPathMask,*
const char * *crlPathMask,*
const char * *dataFilename,*
const char * *currentCmsFile,*
UInt32 * *resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN]);*

Назначение:

Проверка ЭП файла.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени

certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
dataFilename	in	Имя файла с данными, если это отсоединенная ЭП или NULL если присоединенная ЭП
currentCmsFile	in	Имя файла CMS с проверяемыми ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_MEM	Недостаточно памяти
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана

ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи файла CMS для присоединенной или отсоединенной ЭП. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.42. *Проверка ЭП файла*

Int32 DECL cr_CheckFile2(**H_TIMER** *timer,*
**const char ***_certPathMask,_
**const char ***_crlPathMask,_
**const char ***_dataFilename,_
**const char ***_currentCmsFile,_
**UInt32 ***_resultNumOfSign,_
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN],*
H_CTX **h_ctx);*

Назначение:

Проверка ЭП файла.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
dataFilename	in	Имя файла с данными, если это отсоединенная ЭП или NULL если присоединенная ЭП
currentCmsFile	in	Имя файла CMS с проверяемыми ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
h_ctx	out	Контекст хранения результирующих сертификатов (требует последующего удаления путем cr_FreeContext)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_MEM	Недостаточно памяти
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_OPEN_FILE	Ошибка открытия файла

ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем

ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи файла CMS для присоединенной или отсоединенной ЭП. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.43. *Проверка ЭП значения хеш-функции*

Int32 DECL cr_CheckHash(**H_TIMER** *timer,*
const char * *certPathMask,*
const char * *crlPathMask,*
H_DIGEST *hash,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt32 * *resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN]);*

Назначение:

Проверка ЭП хеша.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
hash	in	Контекст хеш-функции для проверяемых данных в случае отсоединенной ЭП
encodedCurrentCms	in	Буфер CMS с проверяемыми ЭП, в случае отсоединенной ЭП
encodedCurrentCmsLength	in	Длина буфера CMS с проверяемыми ЭП, в случае отсоединенной ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа

ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS

ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи значения хеш-функции для отсоединенной ЭП, заранее рассчитанной функциями СКЗИ. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.44. *Проверка ЭП значения хеш-функции*

Int32 DECL cr_CheckHash2(**H_TIMER** *timer,*
const char * *certPathMask,*
const char * *crlPathMask,*
H_DIGEST *hash,*
UInt8 * *encodedCurrentCms,*
UInt32 *encodedCurrentCmsLength,*
UInt32 * *resultNumOfSign,*
Int32 *resultSign[MAX_NUMOF_SIGN],*
H_STORAGE *resultStorage[MAX_NUMOF_SIGN],*
H_CTX **h_ctx);*

Назначение:

Проверка ЭП хеша.

Параметры:

timer	in	Заполненный контекст таймера или NULL для проверки на текущий момент времени
certPathMask	in	Путь к файлам сертификатов, возможно использование маски *.cer
crlPathMask	in	Путь к файлам стоп-листов, возможно использование маски *.crl
hash	in	Контекст хеш-функции для проверяемых данных в случае отсоединенной ЭП
encodedCurrentCms	in	Буфер CMS с проверяемыми ЭП, в случае отсоединенной ЭП
encodedCurrentCmsLength	in	Длина буфера CMS с проверяемыми ЭП, в случае отсоединенной ЭП
resultNumOfSign	out	Количество проверенных ЭП
resultSign[MAX_NUMOF_SIGN]	out	Результат проверки ЭП, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
resultStorage[MAX_NUMOF_SIGN]	out	Индекс сертификатов подписантов в хранилище, длина массива равна количеству проверенных ЭП, но не более MAX_NUMOF_SIGN
h_ctx	out	Контекст хранения результирующих сертификатов (требует последующего удаления путем cr_FreeContext)

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_INVALID_TIME	Неверное время
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_OPEN_FILE	Ошибка открытия файла
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_READ_FILE	Ошибка чтения файла
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано

ERR_NOT_SIGNED_DATA_TYPE	Не найдены подписанные данные
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NO_SIGN	ЭП не найдена или файл не подписан
ERR_CMS_DONT_HAVE_DATA	Данные в CMS не обнаружены
ERR_HASH_NOT_EQUAL	Хеш данных не соответствует хешу в CMS
ERR_CMS_NO_SIGNER_INFO	Не найдена ЭП с требуемым номером
ERR_HASH_ATTRIBUTES	Ошибка в атрибутах поля хеш

Описание функции:

Проверка электронной подписи значения хеш-функции для отсоединенной ЭП, заранее рассчитанной функциями СКЗИ. Для успешной проверки корневой сертификат должен находиться в конфигурационном файле в разделе добавленные корневые сертификаты.

7.2.45. *Проверка ЭП сертификата из хранилища*

Int32 DECL cr_CheckCertificateByStorage(**H_CTX** *ctx,*
H_STORAGE *itemN,*
H_STORAGE * *nextN);*

Назначение:

Проверка ЭП сертификата из хранилища.

Параметры:

ctx	in	Контекст библиотеки
itemN	in	Индекс сертификата в хранилище
nextN	out	Индекс следующего сертификата в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_MEM	Недостаточно памяти
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_TEMPLATE	Ошибка шаблона
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1

ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_BAD_PEM	Ошибка формата данных PEM

Описание функции:

Функции проверки ЭП сертификата в случае успешной проверки возвращает следующий элемент цепочки сертификатов, сертификат издателя. Для получения следующего элемента (индекса в хранилище) следует использовать функцию cr_NextByStorage. Для получения индекса корневого сертификата следует использовать функцию cr_GetRootByStorage.

7.2.46. *Проверка ЭП сертификата из буфера по хранилищу*

Int32 DECL cr_CheckCertificate(**H_CTX** *ctx,*
UInt8* *binaryCert,*
UInt32 *binaryCertLength,*
H_STORAGE * *nextN);*

Назначение:

Проверка ЭП сертификата из буфера. Код ошибки равен ERR_OK в случае успешной проверки.

Параметры:

ctx	in	Контекст библиотеки
binaryCert	in	Буфер с сертификатом в формате ASN.1 или PEM
binaryCertLength	in	Длина буфера с сертификатом
nextN	out	Индекс следующего сертификата в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_MEM	Недостаточно памяти
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе

ERR_NOT_FOUND	Поиск не оказался результативным
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_BAD_PEM	Ошибка формата данных PEM

Описание функции:

Функции проверки ЭП сертификата в случае успешной проверки возвращает следующий элемент цепочки сертификатов, сертификат издателя. Для получения следующего элемента (индекса в хранилище) следует использовать функцию cr_NextByStorage. Для получения индекса корневого сертификата следует использовать функцию cr_GetRootByStorage.

7.2.47. *Проверка ЭП запроса на сертификат*

Int32 DECL cr_CheckPKCS10(**H_CTX** *ctxX,*
UInt8* *pkcs,*
UInt32 *pkcsLength);*

Назначение:

Проверка ЭП запроса на сертификат

Параметры:

ctxX	in	Контекст библиотеки
pkcs	in	Буфер проверяемого запроса на сертификат
pkcsLength	in	Длина буфера проверяемого запроса на сертификат

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана

ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_BUFFER_IN_STACK_TO_SMALL	Внутренняя ошибка буфера, работа невозможна
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Запрос на сертификат является самоподписанным. ЭП запроса на сертификата проверяется с помощью открытого ключа, находящегося в запросе на сертификат.

7.2.48. *Проверка ЭП списка отозванных сертификатов по хранилищу*

Int32 DECL cr_CheckCrl(**H_CTX** *ctx,*
UInt8* *binary_crl,*
UInt32 *crlLength,*
H_STORAGE * *nextN);*

Назначение:

Проверка ЭП списка отозванных сертификатов по хранилищу.

Параметры:

ctx	in	Контекст библиотеки
binary_crl	in	Буфер проверяемого стоп-листа

crlLength	in	Длина буфера проверяемого стоп-листа
nextN	out	В случае успешной проверки - индекс сертификата подписанта для списка отозванных сертификатов в хранилище

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_MEM	Недостаточно памяти
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CRL_TIME_ELAPSED	Истек срок действия стоп-листа
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_INVALID_TIME	Неверное время
ERR_CERTIFICATE_NOT_FOUND	Не найден следующий сертификат в цепочке
ERR_NOT_ISSUER_CERTIFICATE	Сертификат выпущен другим издателем
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_CHOICE_TYPE	Ошибка определения типа

ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_LOAD_KEY	Ошибка загрузки ключа
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_SIGN	ЭП не верна
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CERTIFICATE_NOT_ROOT	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_CERT_IN_CRL	Ошибка сертификат в стоп-листе
ERR_BAD_PEM	Ошибка формата данных PEM

Описание функции:

Можно либо добавить стоп-лист в хранилище с помощью cr_AddCrl, тогда он будет проверен автоматически, это приведет к тому, что в случае успешного добавления все сертификаты, содержащиеся в хранилище и одновременно в стоп-листе, получат статус ошибки ERR_CERT_IN_CRL. Либо можно проверить стоп-лист с помощью данной функции, тогда сертификаты, содержащиеся в хранилище и одновременно в стоп-листе не получат статус ошибки ERR_CERT_IN_CRL. Будет получен только результат проверки стоп-листа.

7.2.49. *Освобождение контекста пути доступа к ключу*

Int32 DECL cr_FreeCarrier(**CARRIER_TYPE** *carrier*);

Назначение:

Освобождение контекста пути доступа к ключу.

Параметры:

carrier	in	Контекст пути доступа к закрытому ключу

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

Описание функции:

Освобождает контекст пути доступа к закрытому ключу.

7.3. Функции для реализации шифрования в соответствии со стандартом X.509

7.3.1. *Инициализация контекста операции зашифрования*

Int32 DECL cr_EncryptInit(**H_CTX** *ctxX,*
UInt32 *numofRecipient,*
UInt8 * *recipientBinaryCert,*
UInt32 * *recipientBinaryCertLength,*
H_CTX_ENCR* *e_ctx);*

Назначение:

Инициализация контекста операции зашифрования.

Параметры:

ctxX	in	Контекст библиотеки
numofRecipient	in	Количество сертификатов абонента
recipientBinaryCert	in	Буфер сертификатов абонентов, идущие подряд
recipientBinaryCertLength	in	Длины буферов сертификатов абонентов в количестве numofRecipient штук
e_ctx	out	Контекст операции шифрования

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией

Описание функции:

Начало работы с контекстом операции зашифрования.

7.3.2. *Накопление данных в контексте операции зашифрования*

Int32 DECL cr_EncryptPutData(**H_CTX_ENCR** *e_ctx,*
UInt8* *data_part,*
UInt32 *dataLength);*

Назначение:

Накопление данных для зашифрования (поддержка потокового режима).

Параметры:

e_ctx	in	Контекст операции шифрования
data_part	in	Буфер с зашифровываемыми данными
dataLength	in	Длина буфера с зашифровываемыми данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Накапливает шифруемые данные.

7.3.3. ***Выполнение шифрования блока данных в контексте операции зашифрования***

Int32 DECL cr_Encrypt(**H_CTX_ENCR** *e_ctx,*
UInt8* *cmsEncryptedData,*
UInt32 * *cmsEncryptedDataLength);*

Назначение:

Зашифровать конечный блок данных (целиком).

Параметры:

e_ctx	in	Контекст операции шифрования
cmsEncryptedData	out	Буфер результирующей CMS с зашифрованными данными
cmsEncryptedDataLength	in/out	Вход - максимально возможный размер буфера результирующей CMS, выход - длина буфера результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init

ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти
ERR_INVALID_TIME	Неверное время
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ
ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CHOICE_TYPE	Ошибка определения типа

Описание функции:

Зашифровывает накопленные данные и формирует результирующий CMS.

7.3.4. ***Закрытие контекста операции зашифрования***

Int32 DECL cr_EncryptClose(**H_CTX_ENCR** *e_ctx);*

Назначение:

Закрытие контекста операции зашифрования.

Параметры:

e_ctx	in	Контекст операции шифрования

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Завершение работы с контекстом операции зашифрования.

7.3.5. ***Зашифрование области памяти***

Int32 DECL cr_EncryptBuffer(**H_CTX** *ctxX,*
UInt8* *data,*
UInt32 *dataLength,*
UInt32 *numofRecipient,*
UInt8 * *recipientBinaryCert,*
UInt32 * *recipientBinaryCertLength,*
UInt8* *cmsEncryptedData,*
UInt32 * *cmsEncryptedDataLength);*

Назначение:

Зашифрование области памяти.

Параметры:

ctxX	in	Контекст библиотеки
data	in	Буфер с зашифровываемыми данными
dataLength	in	Длина буфера с зашифровываемыми данными
numofRecipient	in	Количество сертификатов абонентов
recipientBinaryCert	in	Буфер сертификатов абонентов, идущие подряд
recipientBinaryCertLength	in	Длины буферов сертификатов абонентов в количестве numofRecipient штук

cmsEncryptedData	out	Буфер результирующей CMS
cmsEncryptedDataLength	in/out	Вход - максимально возможный размер буфера результирующей CMS, выход - длина буфера результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_BAD_SELFTEST	Внутренняя ошибка теста криптофункций, работа невозможна
ERR_MEM	Недостаточно памяти
ERR_UNSUPPORTED_ALGORITHM	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_INVALID_TIME	Неверное время
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_BAD_PEM	Ошибка формата данных PEM
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CERTIFICATE_TIME_ELAPSED	Истек срок действия сертификата
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_UNKNOWN_OID	Неизвестный OID алгоритма
ERR_NOT_USED_DSCH	Требуется ключ ПДСЧ, но он не был загружен ранее
ERR_LOAD_GRN_DLL	Ошибка загрузки библиотеки
ERR_STOP	Работа остановлена пользователем (например, был нажат ESC или Ctrl-C)
ERR_DSCH	Ошибка ПДСЧ

ERR_BAD_CRYP	Ошибка шифрования
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_CHOICE_TYPE	Ошибка определения типа

Описание функции:

Зашифровывает переданные данные, формирует результирующую CMS.

7.3.6. *Инициализация контекста операции расшифрования*

Int32 DECL cr_DecryptInit(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
H_CTX_DECR* *d_ctx);*

Назначение:

Инициализация контекста операции расшифрования.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
d_ctx	out	Контекст операции расшифрования

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MEM	Недостаточно памяти

Описание функции:

Начало работы с контекстом операции расшифрования.

7.3.7. *Добавление в контекст операции расшифрования существующей CMS*

Int32 DECL cr_DecryptPutEnvelopedCms(**H_CTX_DECR** *d_ctx,*
UInt8* *cmsEncryptedData,*
UInt32 *cmsEncryptedDataLength);*

Назначение:

Добавить в контекст операции CMS (поддержка потокового режима).

Параметры:

d_ctx	in	Контекст операции расшифрования
cmsEncryptedData	in	Буфер CMS с зашифрованными данными
cmsEncryptedDataLength	in	Длина буфера CMS с зашифрованными данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных

Описание функции:

Добавляет в контекст операции CMS с зашифрованными данными.

7.3.8. ***Выполнение расшифрования блока данных в контексте операции расшифрования***

Int32 DECL cr_Decrypt(**H_CTX_DECR** *d_ctx,*
UInt8* *decryptedData,*
UInt32 * *decryptedDataLength);*

Назначение:

Расшифровать конечный блок данных (целиком).

Параметры:

d_ctx	in	Контекст операции расшифрования
decryptedData	out	Буфер с расшифрованными данными
decryptedDataLength	in/out	Вход - максимально возможный размер буфера с расшифрованными данными, выход - длина буфера с расшифрованными данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

ERR_NOT_ENCRYPTED_DATA_TYPE	Данные в CMS не зашифрованы
ERR_MEM	Недостаточно памяти
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Расшифровывает накопленную CMS с данными, возвращает расшифрованные данные.

7.3.9. ***Закрытие контекста операции расшифрования***

Int32 DECL cr_DecryptClose(**H_CTX_DECR** *d_ctx);*

Назначение:

Закрытие контекста операции расшифрования.

Параметры:

d_ctx	in	Контекст операции расшифрования

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров

Описание функции:

Завершение работы с контекстом операции расшифрования.

7.3.10. *Расшифрование области памяти*

Int32 DECL cr_DecryptBuffer(**H_CTX** *ctx,*
H_KEYPAIR *keypair,*
UInt8* *cmsEncryptedData,*
UInt32 *cmsEncryptedDataLength,*
UInt8* *decryptedData,*
UInt32* *decryptedDataLength);*

Назначение:

Расшифрование области памяти.

Параметры:

ctx	in	Контекст библиотеки
keypair	in	Контекст ключевой пары
cmsEncryptedData	in	Буфер CMS с зашифрованными данными
cmsEncryptedDataLength	in	Длина буфера CMS с зашифрованными данными
decryptedData	out	Буфер с расшифрованными данными
decryptedDataLength	in/out	Вход - максимально возможный размер буфера с расшифрованными данными, выход - длина буфера с расшифрованными данными

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

ERR_INIT	Библиотека не проинициализирована при помощи cr_init
ERR_BAD_HANDLE	Неверный контекст библиотеки, например, контекст был закрыт ранее
ERR_BAD_PARAM	Неправильные параметры или нулевое значение параметров
ERR_NOT_ENCRYPTED_DATA_TYPE	Данные в CMS не зашифрованы
ERR_MEM	Недостаточно памяти
ERR_BAD_LEN	Длина превышает допустимый размер
ERR_BAD_USER	Ошибка загрузки ключевых данных пользователя
ERR_NOT_FOUND	Поиск не оказался результативным
ERR_STRUCT_VALUE_NOT_SET	Значение структуры не задано
ERR_MORE_DATA	Выходной буфер имеет размер меньше, чем требуется для размещения данных
ERR_BAD_ASN1_OBJECT	Нарушена структура ASN1 объекта
ERR_TEMPLATE	Ошибка шаблона
ERR_INVALID_UNDEFINED_LENGTH	Длина объекта не задана
ERR_DECODE	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_UNKNOWN_ASN1_TYPE	Неизвестный тип ASN1 объекта
ERR_LONG_LENGTH	Длина закодированного OID имеет неправильный размер
ERR_TAG_MISMATCH	Ошибка тега ASN1
ERR_CHOICE_TYPE	Ошибка определения типа
ERR_DIFFERENT	Сравниваемые объекты различны
ERR_OPEN_FILE	Ошибка открытия файла
ERR_WRITE_FILE	Ошибка записи файла
ERR_HANDLE_TYPE	Неверный тип контекста библиотеки, например, используется неподходящий контекст
ERR_BAD_CRYPTOCORE	Внутренняя ошибка работы криптоядра, работа невозможна

Описание функции:

Для расшифрования области памяти используется закрытый ключ в контексте ключевой пары.

7.4. Функции центра сертификации

7.4.1. *Создание подписанного CMS включающего PKCS10 запрос*

Int32 DECL ConfirmCertReq(**H_KEYPAIR** *Akeypair,*
UInt8 **encCertReq,*
UInt32 *encCertReqLength,*
UInt8 **encCertificate,*
UInt32 *encCertificateLength,*
UInt8 ***encCertChain,*
UInt32 **encCertChainLengths,*
UInt32 *encCertChainLength,*
Int32 *validityPeriod,*
UInt8 **encConfirmedCertReq,*
UInt32 **encConfirmedCertReqLength);*

Назначение:

Функция создаёт подписанный администратором ЦС CMS с PKCS10 запросом внутри.

Параметры:

Akeypair	in	Дескриптор ключевой пары администратора ЦС
encCertReq	in	Закодированный ASN.1 запрос на сертификат
encCertReqLength	in	Длина закодированного ASN.1 запроса на сертификат
encCertificate	in	Закодированный ASN.1 сертификат администратора ЦС
encCertificateLength	in	Длина закодированного ASN.1 сертификата администратора ЦС
encCertChain	in	Закодированная ASN.1 цепочка сертификата администратора ЦС
encCertChainLengths	in	Длина закодированный ASN.1 цепочки сертификата администратора ЦС
encCertChainLength	in	Количество сертификатов в цепочке
validityPeriod	in	"Срок действия сертификата", "1.2.643.3.123.4.6"
encConfirmedCertReq	out	Результирующая CMS
encConfirmedCertReqLength	in/out	На входе максимальная длина результирующей CMS, на выходе длина результирующей CMS

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
Значение, отличное от нуля	Согласно Списку ошибок, *см. Приложение №1*

Описание функции:

Функция предназначена для работы в Центре Сертификации. Функция создаёт подписанный администратором ЦС CMS с PKCS10 запросом внутри.

7.4.2. ***Создание подписанного ключом администратора ЦС сертификата на основе запроса***

Int32 DECL IssueCertificate (**H_KEYPAIR** *Akeypair,*
UInt8 * *encodedSignedCertReq,*
UInt32 *encodedSignedCertReqLength,*
UInt8 **encRaCertificate,*
UInt32 *encRaCertificateLength,*
UInt8 ***encCertChain,*
UInt32 **encCertChainLengths,*
UInt32 *encCertChainLength,*
UInt8 **serialNumber,*
UInt32 *serialNumberLength,*
Int32 *defaultValidityPeriod,*
ExtensionAction **extensionActions,*
UInt32 *extensionActionsCount,*
UInt8 **fixSubjectEncodedCertReq,*
UInt32 *fixSubjectEncodedCertReqLength,*
UInt8 **encodedCertificate,*
UInt32 **encodedCertificateLength);*

Назначение:

Функция создаёт сертификат на основе запроса, подписывая сертификат ключом администратора УЦ.

Параметры:

Akeypair	in	Дескриптор ключевой пары администратора ЦС
encodedSignedCertReq	in	Закодированный ASN.1 запрос на сертификат
encodedSignedCertReqLength	in	Длина закодированного ASN.1 запроса на сертификат
encRaCertificate	in	Закодированный ASN.1 сертификат RA
encRaCertificateLength	in	Длина закодированного ASN.1 сертификат RA
encCertChain	in	Закодированная ASN.1 цепочка сертификата администратора ЦС
encCertChainLengths	in	Длина закодированный ASN.1 цепочки сертификата администратора ЦС
encCertChainLength	in	Количество сертификатов в цепочке
serialNumber	in	Серийный номер
serialNumberLength	in	Длина серийного номера
defaultValidityPeriod	in	Срок действия
extensionActions	in	Дополнительные действия

extensionActionsCount	in	Количество дополнительных действий
fixSubjectEncodedCertReq	in	Субъект дополнительных действий
fixSubjectEncodedCertReqLength	in	Длина субъекта дополнительных действий
encodedCertificate	in	Закодированный ASN.1 результирующий сертификат
encodedCertificateLength	in/out	Длина закодированного ASN.1 результирующего сертификата

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
Значение, отличное от нуля	Согласно Списку ошибок, *см. Приложение №1*

Описание функции:

Функция предназначена для работы в Центре Сертификации. Функция создаёт сертификат на основе запроса, подписывая сертификат ключом администратора УЦ.

7.4.3. ***Создание ключевой пары администратора ЦС на основе ключа в формате БиКрипт***

Int32 DECL CreateKeyPairFromBicrKey (**H_CTX** *Actx*,
H_USER *h_user,*
const char **Filename,*
UInt8 **buffer,*
UInt32 *bufferLength,*
UInt32 *CertificateInChainToAdd,*
H_KEYPAIR* *Akeypair*);

Назначение:

Функция создаёт ключевую пару на основе секретного ключа в формате БиКрипт и сертификата пользователя.

Параметры:

Actx	in	Контекст библиотеки
h_user	in	Секретный ключ в формате БиКрипт
Filename	in	Имя файла сертификата
buffer	in	Буфер сертификата
bufferLength	in	Длина буфера сертификата
CertificateInChainToAdd	in	Количество сертификатов в цепочке помещаемой в CMS при создании ЭП при помощи этой ключевой парой
Akeypair	out	Ключевая пара

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
Значение, отличное от нуля	Согласно Списку ошибок, *см. Приложение №1*

Описание функции:

Функция предназначена для работы в Центре Сертификации. Функция создаёт ключевую пару на основе секретного ключа и сертификата. При этом проверяется соответствие секретного ключа сертификату.

7.4.4. *Создание корневого сертификата*

Int32 DECL IssueRootCertificate (**H_CTX** *ActxZ,*
H_USER *h_user,*
UInt8 **encodedCertReq,*
UInt32 *encodedCertReqLength,*
UInt8 **serialNumber,*
UInt32 *serialNumberLength,*
UInt32 *defaultValidityPeriod,*
ExtensionAction **extensionActions,*
UInt32 *extensionActionsCount,*
UInt8 **fixSubjectEncodedCertReq,*
UInt32 *fixSubjectEncodedCertReqLength,*
UInt8 **encodedCertificate,*
UInt32 **encodedCertificateLength);*

Назначение:

Функция создаёт корневой сертификат на основе секретного ключа в формате БиКрипт и запроса на сертификат.

Параметры:

ActxZ	in	Контекст библиотеки
h_user	in	Секретный ключ в формате БиКрипт
encodedCertReq	in	Закодированный ASN.1 запрос на сертификат
encodedCertReqLength	in	Длина закодированного ASN.1 запрос на сертификат
serialNumber	in	Серийный номер
serialNumberLength	in	Длина серийного номера
defaultValidityPeriod	in	Срок действия
extensionActions	in	Дополнительные действия
extensionActionsCount	in	Количество дополнительных действий

fixSubjectEncodedCertReq	in	Субъект дополнительных действий
fixSubjectEncodedCertReqLength	in	Длина субъекта дополнительных действий
encodedCertificate	in	Закодированный ASN.1 результирующий корневой сертификат
encodedCertificateLength	in/out	Длина закодированного ASN.1 результирующего корневого сертификата

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено
Значение, отличное от нуля	Согласно списку ошибок, *см.Приложение №1*

Описание функции:

Функция предназначена для работы в Центре Сертификации. Функция создаёт корневой сертификат на основе запроса, подписывая сертификат секретным ключом в формате БиКрипт.

7.4.5. ***Создание подписанного ключом администратора ЦС списка отозванных сертификатов***

Int32 DECL CreateCrl(**H_KEYPAIR** *Akeypair,*
UInt8 **encodedCrl,*
UInt32 *encodedCrlLength,*
UInt8 **encodedSignedCrl,*
UInt32 **encodedSignedCrlLength);*

Назначение:

Функция создаёт СОС, подписывая его ключом администратора УЦ.

Параметры:

Akeypair	in	Дескриптор ключевой пары администратора ЦС
encodedCrl	in	Закодированный Crl
encodedCrlLength	in	Длина закодированного Crl
encodedSignedCrl	in	Подписанный результирующий Crl
encodedSignedCrlLength	in/out	На входе максимальная длина, на выходе длина подписанного результирующего Crl

Возвращаемое значение:

Функция возвращает код ошибки:

ERR_OK	Ошибок не обнаружено

Значение, отличное от нуля	Согласно списку ошибок, *см. Приложение №1*

Описание функции:

Функция предназначена для работы в Центре Сертификации. Функция создаёт COC, подписывая его ключом администратора УЦ.

8 Используемые дескрипторы

H_INIT	дескриптор, характеризующий инициализацию библиотеки
H_USER	закрытый ключ пользователя и его идентификатор
H_PKEY	открытый ключ пользователя и его идентификатор
H_PKBASE	справочник открытых ключей
H_HASH	промежуточные значения хеш-функции
H_CTX	Контекст библиотеки
H_TIMER	Контекст времени
H_DIGEST	Контекст хеш-функции
H_KEYPAIR	Контекст ключевой пары
H_STORAGE	Индекс сертификата в хранилище
H_CTX_CHECK	Контекст операции проверки ЭП
H_CTX_SIGN	Контекст операции формирования ЭП
H_CTX_ENCR	Контекст операции зашифрования
H_CTX_DECR	Контекст операции расшифрования
H_TLS	Контекст операции TLS
CARRIER_TYPE	Контекст пути доступа закрытого ключа (ТМ, флеш-диск с путём)

9 Используемые константы

CR_HDR_SIZE = 60 минимальный размер криптозаголовка в байтах, при зашифровании каждые последующие 4 мегабайта данных увеличивают криптозаголовок в пропорциональное количество раз.

Приложение №1

Справочник кодов ошибок СКЗИ "Бикрипт 5.0"

Название ошибки	Код ошибки	Описание ошибки
ERR_MEM	1	Недостаточно памяти
ERR_BAD_SIGN	2	Подпись неверна
ERR_BAD_LEN	3	Длина буфера неверна
ERR_BAD_USER	4	Номер пользователя неверен
ERR_CRYDRV	5	Ошибка средств шифрования
ERR_INIT_CRYMASK	6	Ошибка расшифрования мастер-ключа
ERR_NOT_SUPPORTED	8	Функция не поддерживается в данной версии или в данном исполнении
ERR_CRC_SK_FILE	9	Ошибка контрольной суммы файла с закрытым ключом
ERR_NO_SIGN	11	Нет подписи
ERR_OPEN_FILE	12	Ошибка открытия файла
ERR_OPEN_PUB	14	Ошибка открытия файла с открытым ключом
ERR_TOO_MANY	16	Слишком много носителей в USB устройстве, поддерживается работа только с одним
ERR_READ_MK_FILE	18	Ошибка чтения файла с мастер-ключом
ERR_SIGN_NO_REG	19	Идентификатор подписи не зарегистрирован в БОК
ERR_BAD_SELFTEST	20	внутренние тесты библиотеки проведены с ошибкой
ERR_GK_READ	21	Ошибка чтения главного ключа
ERR_UZ_READ	22	Ошибка чтения узла замены
ERR_GKUZ_PSW	24	Главный ключ требует ввода пароля
ERR_DSCH	25	Не найден датчик ДСЧ
ERR_CRC_TM	26	Ошибка контрольной суммы при чтении ТМ
ERR_LOAD_GRN_DLL	27	Ошибка загрузки библиотеки grn.dll или libbiogrn.dylib
ERR_STOP	28	Остановлено пользователем
ERR_TMDRV_NOT_FOUND	29	Нет драйвера ТМ
ERR_NO_TM_ATTACHED	30	Не приложена ТМ к съемнику
ERR_READ_TM	31	Ошибка чтения ТМ
ERR_BAD_PARAM	32	Ошибка в параметрах функции

Название ошибки	Код ошибки	Описание ошибки
ERR_BAD_HANDLE	33	Ошибка дескриптора (например, происходит обращение к закрытому ранее дескриптору или вместо валидного дескриптора используется случайное значение)
ERR_HANDLE_TYPE	34	Неправильный тип дескриптора (например, вместо типа H_USER в соответствующий параметр функции передается дескриптор типа H_PKEY)
ERR_WRITE_TM	35	Ошибка записи ТМ
ERR_BAD_CRYPTOCORE	36	Внутренняя ошибка работы криптоядра, работа невозможна
ERR_INIT	39	Ошибка инициализации библиотеки, не был вызван cr_init
ERR_LOAD_KEY	40	Ошибка загрузки ключа
ERR_NO_CRYP	43	Буфер не был зашифрован
ERR_BAD_CRYP	44	Ошибка расшифрования буфера
ERR_FILE_KEY	45	Ошибка файлового ключа
ERR_READ_FILE	46	Ошибка чтения файла
ERR_WRITE_FILE	47	Ошибка записи файла
ERR_COMPRESS	48	Ошибка компрессии
ERR_MORE_DATA	49	Ошибка - длина буфера недостаточна
ERR_UNKNOWN_ASN1_TYPE	300	Неизвестный тип ASN1 объекта
ERR_DECODE	302	Ошибка декодирования, возможно, испорчены данные или неверен формат
ERR_LONG_LENGTH	304	Длина закодированного OID имеет размер более 4 байт
ERR_TEMPLATE	305	Ошибка шаблона
ERR_CHOICE_TYPE	306	Ошибка определения типа
ERR_BAD_ASN1_OBJECT	307	Нарушена структура ASN1 объекта
ERR_UNKNOWN_OID	308	Неизвестный OID алгоритма
ERR_UNKNOWN_ID	309	Неизвестный ID алгоритма
ERR_STRUCT_VALUE_NOT_SET	311	Значение структуры не задано
ERR_INVALID_TIME	313	Неверное время
ERR_NOT_SIGNED_DATA_TYPE	314	Не найдены подписанные данные
ERR_CMS_NO_SIGNER_INFO	315	Не найдена ЭП с требуемым номером
ERR_CMS_DONT_HAVE_DATA	316	Данные в CMS не обнаружены

Название ошибки	Код ошибки	Описание ошибки
ERR_UNSUPPORTED_ALGORITHM	317	Алгоритм несовместим с текущей криптографической операцией
ERR_BAD_PEM	319	Ошибка формата данных PEM
ERR_DIFFERENT	320	Сравниваемые объекты различны
ERR_NOT_ISSUER_CERTIFICATE	321	Сертификат выпущен другим издателем
ERR_NOT_FOUND	322	Поиск не оказался результативным
ERR_HASH_ATTRIBUTES	323	Ошибка в атрибутах поля хеш
ERR_HASH_NOT_EQUAL	324	Хеш данных не соответствует хешу в CMS
ERR_BAD_CERTIFICATES_COUNT	325	Построена слишком короткая цепочка, число сертификатов не соответствует реальной длине
ERR_CERTIFICATE_NOT_FOUND	326	Не найден следующий сертификат в цепочке
ERR_CERTIFICATE_TIME_ELAPSED	327	Истек срок действия сертификата
ERR_CERTIFICATE_NOT_ROOT	328	Корневой сертификат не найден в списке доверенных сертификатов в конфигурационном файле
ERR_NOT_ENCRYPTED_DATA_TYPE	331	Данные в CMS не зашифрованы
ERR_CERT_IN_CRL	332	Ошибка сертификат в стоп-листе
ERR_CRL_BAD_SIGN	333	ЭП стоп-листа неверна
ERR_CRL_TIME_ELAPSED	334	Истек срок действия стоп-листа
ERR_INVALID_UNDEFINED_LENGTH	335	Длина объекта не задана
ERR_TAG_MISMATCH	336	Ошибка тега ASN1
ERR_BUFFER_IN_STACK_TO_SMALL	338	Внутренняя ошибка буфера, работа невозможна
ERR_CA_CONSTRAINTS_UNSATISFACT	339	Ошибка ЦС – сертификат ключа имеет ограниченные права выпуска других сертификатов
ERR_IMPROPER_CERTIFICATE	340	Ошибка ЦС – неправильный сертификат
ERR_PARAMETERS_ABSENT	342	Ошибка ЦС – параметр отсутствует
ERR_PATH_LEN_CONSTRAINT	343	Ошибка ЦС - сертификат ключа имеет ограниченную длину цепочки выпуска других сертификатов
ERR_ALREADY	344	Уже присутствует
ERR_UNSUPPORTED	2222	Не поддерживаются в данной версии
ERR_NOT_USED_TMDRV	11110	Драйвер ТМ не найден, но осуществляется попытка его использования

Название ошибки	Код ошибки	Описание ошибки
ERR_NOT_USED_DSCH	11111	Датчик ДСЧ не проинициализирован, но осуществляется попытка его использования
ERR_NOT_USED_GKUZ	11112	Главный ключ не введен, но осуществляется попытка его использования

Приложение №2

Список функций стандартной версии и версии без шифровальных функций ("+" - функция входит в указанную версию, "-" - не входит).

Имя функции	Краткое описание	Стандартная версия	Версия без шифрования (SIGN_ONLY)
cr_init	Инициализация библиотеки	+	+
cr_uninit	Деинициализации библиотеки	+	+
cr_init_random	Инициализация программного датчика случайных чисел	+	+
cr_set_procent_callback	Инициализация функции вывода процента готовности	+	+
cr_get_version_info	Получение информации об используемой библиотеке	+	+
cr_get_tm_number	Получение номера ТМ - идентификатора, приложенного к считывателю	+	+
cr_set_param	Установка параметров библиотеки	+	+
cr_get_param	Получение параметров библиотеки	+	+
cr_init_prnd	Инициализация ПДСЧ	+	+
cr_write_prnd	Запись данных ПДСЧ	+	+
cr_hash_open	Инициализация вычислений значений хеш-функции	+	+
cr_hash_calc	Вычисление значения хеш-функции для части блока данных	+	+
cr_hash_return	Завершение вычисления значения хеш-функции для блока данных	+	+
cr_hash_close	Деинициализация вычислений значений хеш-функции	+	+
cr_hash_set	Установка значения хеш-функции	+	+
cr_hash_dup	Копирование дескриптора хеш-функции	+	+
cr_gen_masterkey	Генерация мастер-ключа	+	-
cr_install_masterkey	Установка мастер-ключа	+	-
cr_change_masterkey	Перешифрование мастер-ключа на новом главном ключе	+	-
cr_gen_onetime_keypair	Генерация временной пары ключей ЭП	+	+
cr_gen_elgkey	Генерация ключей ЭП с паролем или привязкой к ТМ	+	+
cr_gen_keypair	Генерация ключей ЭП в файл с паролем	+	+
cr_gen_elgkey_mk	Генерация ключей ЭП с маской - мастер-ключом	+	-
cr_gen_keypair_mk	Генерация ключей ЭП в файл с маской - мастер-ключом	+	-
cr_load_elgkey	Загрузка закрытого ключа ЭП с паролем или привязкой к ТМ	+	+
cr_read_skey	Загрузка из файла закрытого ключа ЭП с паролем	+	+
cr_load_elgkey_mk	Загрузка закрытого ключа ЭП с маской - мастер-ключом	+	-
cr_read_skey_mk	Загрузка из файла закрытого ключа ЭП с мастер-ключом	+	-
cr_elgkey_getid	Получение идентификатора закрытого ключа ЭП	+	+
cr_elgkey_close	Выгрузка закрытого ключа ЭП	+	+
cr_gen_pubkey	Генерация открытого ключа ЭП	+	+
cr_gen_gk	Генерация Главного ключа	+	-
cr_gk_copy_tm	Копирование главного ключа в ТМ	+	-

cr_gen_elgkey_ext	Расширенная генерация ключей ЭП с паролем или “привязкой” к ТМ	+	+
cr_gen_elgkey_mk_ext	Расширенная генерация ключей ЭП с маской - мастер-ключом	+	-
cr_change_elgkey	Замена закрытого ключа ЭП с паролем или “привязкой” к ТМ	+	+
cr_pkbase_open	Открытие справочника	+	+
cr_pkbase_find	Проверка наличия открытого ключа ЭП в справочнике	+	+
cr_pkbase_add	Добавление/замена открытого ключа ЭП в справочник(е)	+	+
cr_pkbase_remove	Удаление открытого ключа ЭП из справочника	+	+
cr_pkbase_save	Сохранение справочника	+	+
cr_pkbase_findfirst	Выбор первого ключа из справочника открытых ключей	+	+
cr_pkbase_findnext	Выбор следующего ключа из справочника открытых ключей	+	+
cr_pkbase_close	Закрытие справочника открытых ключей	+	+
cr_write_tm	Запись в ТМ значения хеш-функции	+	+
cr_read_tm	Чтение из ТМ значения хеш-функции	+	+
cr_pkey_put	Создание дескриптора открытого ключа	+	+
cr_pkey_load	Загрузка открытого ключа из буфера	+	+
cr_gen_pubkey	Создание открытого ключа	+	+
cr_pkey_getid	Получение идентификатора открытого ключа	+	+
cr_pkey_getinfo	Экспортирование открытого для вывода на печать	+	+
cr_pkey_close	Освободить дескриптор, характеризующий открытый ключ	+	+
cr_sign_buf	Формирование ЭП для блока памяти	+	+
cr_check_buf	Проверка ЭП для блока памяти	+	+
cr_sign_hash	Формирование ЭП для значения хеш-функции	+	+
cr_check_hash	Проверка ЭП для значения хеш-функции	+	+
cr_sign_file	Формирование ЭП для содержимого файла	+	+
cr_check_file	Проверка ЭП для содержимого файла	+	+
cr_buf_put_sign_struct	Помещение структуры данных подписи в конец буфера	+	+
cr_file_put_sign_struct	Помещение структуры данных подписи в конец файла	+	+
cr_buf_get_sign_struct	Получение структуры данных подписи для буфера	+	+
cr_file_get_sign_struct	Получение структуры данных подписи для файла	+	+
cr_GetNextByStorage	Получение следующего индекса цепочки сертификатов	+	+
cr_GetRootByStorage	Получение корневого сертификата в цепочке сертификатов	+	+
cr_AddCertificate	Добавление сертификата в контекст библиотеки	+	+
cr_AddCertificateAndCrlByPath	Добавление сертификатов и списков отозванных сертификатов из файлов в контекст библиотеки	+	+
cr_AddCrl	Добавление списка отозванных сертификатов в контекст библиотеки	+	+
cr_Check	Проверка ЭП и получение результатов в контексте операции проверки ЭП	+	+
cr_CheckBuffer	Проверка ЭП области памяти	+	+
cr_CheckCertificate	Проверка ЭП сертификата из буфера по хранилищу	+	+

cr_CheckCertificateByStorage	Проверка ЭП сертификата из хранилища	+	+
cr_CheckClose	Закрытие контекста операции проверки ЭП	+	+
cr_CheckCrl	Проверка ЭП списка отозванных сертификатов по хранилищу	+	+
cr_CheckFile	Проверка ЭП файла	+	+
cr_CheckHash	Проверка ЭП значения хеш-функции	+	+
cr_CheckInit	Инициализация контекста операции проверки ЭП	+	+
cr_CheckPKCS10	Проверка ЭП запроса на сертификат	+	+
cr_CheckPutCms	Добавление в контекст операции проверки ЭП существующей CMS	+	+
cr_CheckPutData	Накопление данных для проверки отсоединенной ЭП в контексте операции проверки ЭП	+	+
cr_CheckPutHash	Добавление значения хеш-функции в контекст операции проверки ЭП	+	+
cr_CopyTime	Копирование контекста времени	+	+
cr_CreateContext	Инициализация контекста библиотеки	+	+
cr_CreateTime	Создание контекста времени	+	+
cr_Decrypt	Выполнение расшифрования блока данных в контексте операции расшифрования	+	-
cr_DecryptBuffer	Расшифрование области памяти	+	-
cr_DecryptClose	Закрытие контекста операции расшифрования	+	-
cr_DecryptInit	Инициализация контекста операции расшифрования	+	-
cr_DecryptPutEnvelopedCms	Добавление в контекст операции расшифрования существующей CMS	+	-
cr_DeleteSignBuffer	Удаление ЭП из CMS	+	+
cr_DigestClose	Закрыть контекст вычисления значений хеш-функции	+	+
cr_DigestFinal	Получение значения в контексте вычисления значений хеш-функции	+	+
cr_DigestInit	Инициализация контекста вычисления значений хеш-функции	+	+
cr_DigestUpdate	Добавление порции данных в контекст вычисления значений хеш-функции	+	+
cr_DupContext	Создание копии контекста библиотеки	+	+
cr_Encrypt	Выполнение шифрования блока данных в контексте операции зашифрования	+	-
cr_EncryptBuffer	Зашифрование области памяти	+	-
cr_EncryptClose	Закрытие контекста операции зашифрования	+	-
cr_EncryptInit	Инициализация контекста операции зашифрования	+	-
cr_EncryptPutData	Накопление данных в контексте операции зашифрования	+	-
cr_Finalize	Завершение работы	+	+
cr_FindKeyPair	Поиск ключевой пары по сертификату в контексте библиотеки	+	+
cr_FreeContext	Освобождение контекста библиотеки	+	+
cr_FreeKeyPair	Освобождение ресурсов ключевой пары	+	+
cr_FreeTime	Освобождение контекста времени	+	+
cr_GenerateKeyPair	Создание ключевой пары на основе выделенного имени и атрибутов, кодированных в формате ASN.1	+	+
cr_GenerateKeyPairByQualNames	Создание ключевой пары на основе данных, идентифицирующих владельца сертификата	+	+
cr_CreateCarrier	Создание контекста пути доступа к закрытому ключу	+	+

cr_FreeCarrier	Освобождение контекста пути доступа к закрытому ключу	+	+
cr_GetBufferByStorage	Получение сертификата из хранилища	+	+
cr_GetCertificateTimeCn	Получение времени ЭП и выделенного имени владельца из сертификата в буфере	+	+
cr_GetCertificateTimeCnByStorage	Получение времени ЭП и выделенного имени владельца из сертификата в хранилище	+	+
cr_GetCertificateUsageInfo	Политика сертификата и класс средства ЭП	+	+
cr_GetChainSizeByStorage	Получение длины цепочки сертификатов в хранилище	+	+
cr_GetCmsData	Получение информации об ЭП из CMS в буфере	+	+
cr_GetCmsFileTimeCn	Получение времени ЭП и выделенного имени подписанта CMS в файле	+	+
cr_GetCmsTimeCn	Получение времени ЭП и выделенного имени подписанта CMS в буфере	+	+
cr_GetCrlTimeCn	Получение времени ЭП и выделенного имени создателя списка отозванных сертификатов	+	+
cr_GetDigestParam	Получение параметров контекста вычисления значений хеш-функции	+	+
cr_GetFileData	Получение информации об ЭП из CMS в файле	+	+
cr_GetFilenameByStorage	Получение имени файла сертификата в хранилище	+	+
cr_GetKeyPairInfo	Получение информации о ключевой паре в контексте библиотеки	+	+
cr_GetNumOfKeyPair	Получение количества ключевых пар в контексте библиотеки	+	+
cr_GetPrndCarrier	Получение информации о носителе ключа ПДСЧ в контексте библиотеки	+	+
cr_GetStatusByStorage	Получение статуса сертификата из хранилища	+	+
cr_GetStorageInfo	Получение количества сертификатов и списка отозванных сертификатов в хранилище	+	+
cr_GetTime	Получение времени из контекста	+	+
cr_LoadKeyPair	Загрузка ключевой пары в контексте библиотеки	+	+
cr_LoadPrnd	Инициализация датчика ПДСЧ	+	+
cr_SetCurrentTime	Установка текущего времени для контекста времени	+	+
cr_SetDigestParam	Задание параметров контекста вычисления значений хеш-функции	+	+
cr_SetDigestParamByKeyPair	Задание параметров контекста вычисления значений хеш-функции, согласованных с ключевой парой	+	+
cr_SetTime	Установка заданного времени в контекст	+	+
cr_Sign	Формирование ЭП в контексте операции формирования ЭП	+	+
cr_SignBuffer	Электронная подпись области памяти	+	+
cr_SignClose	Закрытие контекста операции формирования ЭП	+	+
cr_SignFile	Электронная подпись файла	+	+
cr_SignHash	Электронная подпись значения хеш-функции	+	+
cr_SignInit	Инициализация контекста операции формирования ЭП	+	+
cr_SignPutCms	Добавление в контекст операции формирования ЭП существующей CMS	+	+
cr_SignPutData	Накопление данных в контексте операции формирования ЭП	+	+

cr_SignPutHash	Добавление значения хеш-функции в контекст операции формирования ЭП	+	+
ConfirmCertReq	Функция создаёт подписанный CMS с запросом внутри	+	+
IssueCertificate	Функция создаёт сертификат на основе запроса, подписывая сертификат ключом администратора ЦС	+	+
CreateCrl	Функция создаёт CRL, подписывая его ключом администратора ЦС	+	+

Приложение №3

Параметры криптографических алгоритмов

Символ	Десятично	Имя параметра	ГОСТ	OID	RFC
'A'	65	Параметр A	по ГОСТ Р 34.10-2001*	1.2.643.2.2.35.1	id-GostR3410-2001-CryptoPro-A-ParamSet
'B'	66	Параметр B	по ГОСТ Р 34.10-2001*	1.2.643.2.2.35.2	id-GostR3410-2001-CryptoPro-B-ParamSet
'C'	67	Параметр C	по ГОСТ Р 34.10-2001*	1.2.643.2.2.35.3	id-GostR3410-2001-CryptoPro-C-ParamSet
'X'	88	Параметр XchA	по ГОСТ Р 34.10-2001*	1.2.643.2.2.36.0	id-GostR3410-2001-CryptoPro-XchA-ParamSet
'Y'	89	Параметр XchB	по ГОСТ Р 34.10-2001*	1.2.643.2.2.36.1	id-GostR3410-2001-CryptoPro-XchB-ParamSet
'a'	97	Параметр A (хеш по ГОСТ 34.11-2012)	по ГОСТ Р 34.10-2012	1.2.643.2.2.35.1	id-GostR3410-2001-CryptoPro-A-ParamSet
'b'	98	Параметр B (хеш по ГОСТ 34.11-2012)	по ГОСТ Р 34.10-2012	1.2.643.2.2.35.2	id-GostR3410-2001-CryptoPro-B-ParamSet
'c'	99	Параметр C (хеш по ГОСТ 34.11-2012)	по ГОСТ Р 34.10-2012	1.2.643.2.2.35.3	id-GostR3410-2001-CryptoPro-C-ParamSet
'x'	120	Параметр XchA (хеш по ГОСТ 34.11-2012)	по ГОСТ Р 34.10-2012	1.2.643.2.2.36.0	id-GostR3410-2001-CryptoPro-XchA-ParamSet
'y'	121	Параметр XchB (хеш по ГОСТ 34.11-2012)	по ГОСТ Р 34.10-2012	1.2.643.2.2.36.1	id-GostR3410-2001-CryptoPro-XchB-ParamSet
'D'	68	Параметр A (хеш по ГОСТ 34.11-2012, 256 бит)	по ГОСТ Р 34.10-2012	1.2.643.7.1.2.1.1.1	id-tc26-gost-3410-2012-256-paramSetA
'1'	49	Параметр A (хеш по ГОСТ 34.11-2012, 512 бит)	по ГОСТ Р 34.10-2012	1.2.643.7.1.2.1.2.1	id-tc26-gost-3410-12-512-paramSetA
'2'	50	Параметр B (хеш по ГОСТ 34.11-2012, 512 бит)	по ГОСТ Р 34.10-2012	1.2.643.7.1.2.1.2.2	id-tc26-gost-3410-12-512-paramSetB
'3'	51	Параметр C (хеш по ГОСТ 34.11-2012, 512 бит)	по ГОСТ Р 34.10-2012	1.2.643.7.1.2.1.2.3	id-tc26-gost-3410-12-512-paramSetC

*В соответствии с письмом Федеральной службы безопасности Российской Федерации от 07.09.2018 №149/7/6-363 использование схемы ЭП, соответствующей ГОСТ Р 34.10-2001 «Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи», для формирования ЭП возможно до 31.12.2019.

Параметры хеш-функции

Символ	Десятично	Имя параметра	ГОСТ	OID	RFC
'H'	72	Параметр	по ГОСТ 34.11-94**	1.2.643.2.2.30.1	id-GostR3411-94-CryptoProParamSet
'F'	70	Параметр хеш 256 бит	по ГОСТ 34.11-2012	1.2.643.7.1.1.2.2	id-tc26-gost3411-12-256
'G'	71	Параметр хеш 512 бит	по ГОСТ 34.11-2012	1.2.643.7.1.1.2.3	id-tc26-gost3411-12-512

** В соответствии с приказом Федерального агентства по техническому регулированию и метрологии от 07.08.2012 №216-ст ГОСТ Р 34.11-94 отменен с 01.01.2013.

Параметры носителя CARRIER_TYPE

CARRIER_FILE = 0

CARRIER_TM = 1

CARRIER_VPNKEY = 2

Значения констант

BAD_STATUS 0xffff

BAD_H_STORAGE 0xffffffff

Приложение №4

Список отличий между реализациями программной библиотеки для Windows систем и для Unix систем:

1. В версии для Unix функция cr_init_random() недоступна и всегда возвращает ошибку ERR_NOT_SUPPORTED, так как в Unix нет библиотеки GRN.DLL.
2. В версии для Unix функция cr_load_bicr_dll() всегда возвращает ERR_OK, так как в Unix нет библиотеки GRN.DLL и ее не нужно загружать дополнительно.
3. В версии для Unix функция cr_sign_hfile() недоступна, так как нет дескриптора типа HANDLE.
4. Используемая директория для файла prnd.db3 в функции cr_init_prnd:
    a. Для Windows версии если имя файла задать пустым в виде "\0", в этом случае файл будет создан в директории пользователя %APPDATA% с именем prnd.db3
    b. Для Unix версии если имя файла задать пустым в виде "\0" в этом случае файл будет создан во временной директории "/tmp/prnd.db3".
5. Используемая директория для временных файлов:
    a. Для Windows версии задается переменной окружения TEMP.
    b. Для Unix версии всегда соответствует "/tmp/".