127018, Москва, Сущёвский Вал, 18
Телефон: (495) 995 4820
https://CryptoPro.ru
E-mail: info@CryptoPro.ru

Средство Криптографической Защиты Информации

КриптоПро CSP Версия 5.0 R3 KC1 (исполнение 1-Base) Руководство программиста. Модуль Check

ЖТЯИ.00101-03 96 04
Листов 9

2024 г

Содержание

1 Общие сведения

Модуль Check СКЗИ КриптоПро CSP представляет собой набор самостоятельных (не требующих установки базовых модулей КриптоПро CSP) программных компонентов, выполняющих функции расчета хэш-значения и проверки ЭП и предназначенных для эксплуатации под управлением ОС СН Astra Linux SE.

Модуль реализован как динамически и статически подключаемые библиотеки. Статически подключаемые библиотеки предназначены для использования только в составе ядра ОС Astra Linux SE с проведением установленным для указанной ОС порядком работ по исследованиям ОС СН Astra Linux SE.

2 Описание интерфейса

2.1 Функции расчета хэш-значения

Функции, вызываемые для расчета хэш-значения в соответствии с ГОСТ Р 34.11-2012 (ГОСТ 34.11-2018), реализованы в `Stribog.c`. Примеры использования данных функций представлены в тестовом файле `test.c`.

2.1.1 Gost_R34_11_2012_init()

Данная функция предназначена для инициализации контекста хэширования.

```
int Gost_R34_11_2012_init( void* contextBuffer, size_t* contextBufferSize, uint32_t digestLen,
GOST_R34_11_HANDLE* outHandle)
```

`contextBuffer`	Буфер для инициализации контекста хэширования. Если передано значение NULL, то в contextBufferSize будет возвращен необходимый размер буфера и функция вернет код GOST_E_BUFFER_TOO_SMALL.
`contextBufferSize`	Указатель на размер буфера contextBuffer. Если значение данного параметра меньше размера структуры GOST_R34_11_CONTEXT, то функция вернет код GOST_E_BUFFER_TOO_SMALL.
`digestLen`	Размер вычисляемого хэш-значения. Данный параметр должен принимать значения 512 или 256, иначе будет возвращена ошибка GOST_E_INVALID_PARAM.
`outHandle`	Дескриптор контекста хэширования. Является выходным параметром функции. Дескриптор возвращается только в случае успешной работы функции, то есть код возврата GOST_SUCCESS.

2.1.2 Gost_R34_11_get_digest_size()

Данная функция возвращает размер хэш-значения, который будет получен при вычислениях.

```
int Gost_R34_11_get_digest_size(GOST_R34_11_HANDLE handle, size_t* outDigestSize)
```

`handle`	Дескриптор контекста хэширования. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.

outDigestSize	Указатель на размер хэш-значения. В случае нулевого указателя будет возвращен код ошибки GOST_E_INVALID_PARAM. В данном параметре возвращается значение из контекста хэширования hashLen. Данное значение инициализируется при вызове Gost_R34_11_2012_init() значением параметра digestLen.

2.1.3 Gost_R34_11_update()

Данная функция предназначена для добавления данных, от которых будет рассчитано хэш-значение, в контекст хэширования. Функция Gost_R34_11_update() может вызываться последовательно с передачей данных для хэширования по частям. При этом выполняются промежуточные вычисления. Для выполнения финальных вычислений для получения хэш-значения («финализация» контекста) необходимо вызвать функцию Gost_R34_11_final().

```
int Gost_R34_11_update(GOST_R34_11_HANDLE handle, const void* input, size_t inputSize)
```

handle	Дескриптор контекста хэширования. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
input	Указатель на добавляемые данные. В случае если размер inputSize ненулевой, а указатель на данные NULL, то будет возвращена ошибка GOST_E_INVALID_PARAM. При этом производятся промежуточные вычисления хэш-значения в случае, если накопленная длина данных кратна 64 байтам.
inputSize	Размер переданных в input данных.

2.1.4 Gost_R34_11_final()

Данная функция предназначена для выполнения финальных вычислений для получения хэш-значения от переданных ранее данных в вызовах Gost_R34_11_update().

```
int Gost_R34_11_final(GOST_R34_11_HANDLE handle, void* digestBuf, size_t digestBufSize)
```

handle	Дескриптор контекста хэширования. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
digestBuf	Указатель на буфер для хэш-значения. Если передан нулевой указатель, будет возвращена ошибка GOST_E_INVALID_PARAM. Если переданы корректные параметры, то осуществляются финальные операции для получения хэш-значения в соответствии с ГОСТ Р 34.11-2012, которое будет возвращено в данном параметре в случае успеха.
digestBufSize	Размер буфера digestBuf. Осуществляется проверка того, что данное значение не меньше размера хэш-значения (поле hashLen контекста хэширования), иначе возвращается код ошибки GOST_E_BUFFER_TOO_SMALL.

2.1.5 Gost_R34_11_clear()

Данная функция предназначена для обнуления контекста хэширования.

```
void Gost_R34_11_clear(GOST_R34_11_HANDLE handle)
```

handle	Дескриптор контекста хэширования. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Контекст хэширования (структура GOST_R34_11_CONTEXT) инициализируется нулями.

2.1.6 Gost_R34_11_clone()

Данная функция предназначена для копирования контекста хэширования.

```
int Gost_R34_11_clone(GOST_R34_11_HANDLE sourceHandle, void* contextBuffer, size_t*
contextBufferSize, GOST_R34_11_HANDLE* outHandle)
```

sourceHandle	Дескриптор копируемого контекста хэширования. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
contextBuffer	Буфер, в который будет скопирован контекст хэширования. Если передано значение NULL, то в contextBufferSize будет возвращен необходимый размер буфера и функция вернет код GOST_E_BUFFER_TOO_SMALL. В данный буфер копируется контекст хэширования (структура GOST_R34_11_CONTEXT) из sourceHandle.
contextBufferSize	Указатель на размер буфера contextBuffer. Если значение данного параметра меньше размера структуры GOST_R34_11_CONTEXT, то функция вернет код GOST_E_BUFFER_TOO_SMALL.
outHandle	Дескриптор скопированного контекста хэширования contextBuffer.

2.2 Функции проверки подписи

Функции проверки электронной подписи в соответствии с ГОСТ Р 34.10-2012 (ГОСТ 34.10-2018) реализованы в GR3410.c. Примеры использования данных функций представлены в тестовом файле test.c.

2.2.1 Gost_R34_10_2012_init()

Данная функция предназначена для инициализации контекста открытого ключа.

```
int Gost_R34_10_2012_init(void* contextBuffer, size_t* contextBufferSize, GostCurveId curveId,
uint32_t flags, GOST_R34_10_HANDLE* outHandle)
```

contextBuffer	Буфер для инициализации контекста открытого ключа. Если передано значение NULL, то в contextBufferSize будет возвращен необходимый размер буфера и функция вернет код GOST_E_BUFFER_TOO_SMALL.
contextBufferSize	Указатель на размер буфера contextBuffer. Если значение данного параметра меньше размера структуры GOST_R34_10_PUBKEY, то функция вернет код GOST_E_BUFFER_TOO_SMALL.
curveId	Идентификатор эллиптической кривой. По данному идентификатору определяются параметры кривой с помощью функции get_curve(), определенной в EasyElliptic.c. Соответствующими параметрами инициализируется контекст открытого ключа.

flags	Флаги, определяющие тип структуры, используемой для контекста открытого ключа. В случае нулевых флагов используется структура GOST_R34_10_PUBKEY. При указании флага GOST_USE_PUB_KEY_OPT_TABLE_FLAG используется структура GOST_R34_10_PUBKEY_TABLED, которая используется при наличии предвычисленных таблиц для открытого ключа.
outHandle	Дескриптор контекста открытого ключа. Является выходным параметром функции. Дескриптор возвращается только в случае успешной работы функции, то есть код возврата GOST_SUCCESS.

2.2.2 Gost_R34_10_2012_set_public_key()

Данная функция предназначена для установки ключа проверки подписи в контекст.

```
int Gost_R34_10_2012_set_public_key(GOST_R34_10_HANDLE handle, const void* pubKey, size_t
pubKeySize)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
pubKey	Указатель на открытый ключ. Если данный параметр не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Значение открытого ключа устанавливается в контекст как координаты точки эллиптической кривой. При этом проверяется принадлежность данной точки уравнению кривой, в случае ошибки возвращается код ошибки GOST_E_INVALID_PARAM.
pubKeySize	Длина открытого ключа. Осуществляется проверка длины на равенство удвоенному модулю кривой. В случае ошибки будет возвращен код GOST_E_INVALID_PARAM.

2.2.3 Gost_R34_10_2012_verify()

Данная функция предназначена для проверки значения подписи.

```
int Gost_R34_10_2012_verify(GOST_R34_10_HANDLE handle, const void* digest, size_t digestSize,
const void* signature, size_t signatureSize)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
digest	Указатель на хэш-значение, от проверяемых данных. Если передан нулевой указатель, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
digestSize	Длина хэш-значения digest. Осуществляется проверка равенства длины модулю кривой. В случае неуспешной проверки возвращается код ошибки GOST_E_INVALID_PARAM.
signature	Указатель на значение подписи. Если передан нулевой указатель, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Осуществляется проверка значения подписи signature в соответствии с ГОСТ Р 34.10-2012.
signatureSize	Длина подписи signature. Осуществляется проверка длины на равенство удвоенному модулю кривой. В случае ошибки будет возвращен код GOST_E_INVALID_PARAM.

2.2.4 Gost_R34_10_2012_clear()

Данная функция предназначена для зануления контекста открытого ключа.

```
int Gost_R34_10_2012_clear(GOST_R34_10_HANDLE handle)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Контекст хэширования (структура GOST_R34_10_PUBKEY) инициализируется нулями.

2.2.5 Gost_R34_10_2012_get_curve_id()

Данная функция предназначена для получения идентификатора эллиптической кривой.

```
int Gost_R34_10_2012_get_curve_id(GOST_R34_10_HANDLE handle, GostCurveId* outCurveId)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Контекст хэширования (структура GOST_R34_10_PUBKEY) инициализируется нулями.
outCurveId	Указатель на выходной буфер, в который будет скопирован идентификатор эллиптической кривой, используемой в контексте handle. Если передан нулевой указатель, то будет возвращен код ошибки GOST_E_INVALID_PARAM.

2.2.6 Gost_R34_10_2012_clone()

Данная функция предназначена для копирования контекста хэширования.

```
int Gost_R34_10_2012_clone(GOST_R34_10_HANDLE sourceHandle, void* contextBuffer, size_t*
contextBufferSize, GOST_R34_10_HANDLE* outHandle)
```

sourceHandle	Дескриптор копируемого контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
contextBuffer	Буфер, в который будет скопирован контекст открытого ключа. Если передано значение NULL, то в contextBufferSize будет возвращен необходимый размер буфера и функция вернет код GOST_E_BUFFER_TOO_SMALL. В данный буфер копируется контекст открытого ключа (структура GOST_R34_10_PUBKEY) из sourceHandle.
contextBufferSize	Указатель на размер буфера contextBuffer. Если значение данного параметра меньше требуемого размера (поле cbSize структуры GOST_R34_10_PUBKEY), то функция вернет код GOST_E_BUFFER_TOO_SMALL.
outHandle	Дескриптор скопированного контекста открытого ключа contextBuffer.

2.2.7 Gost_R34_10_2012_serialize()

Данная функция предназначена для получения данных, содержащихся в контексте открытого ключа: идентификатор эллиптической кривой, ее координаты и модуль.

```
int Gost_R34_10_2012_serialize(GOST_R34_10_HANDLE handle, void* serializeToBuf, size_t*
serializeToBufSize)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM.
serializeToBuf	Указатель на выходной буфер, в который копируются данные контекста открытого ключа. Если передан нулевой указатель, то будет возвращен необходимый размер буфера в serializeToBufSize. В буфер копируются значения из контекста открытого ключа (структура GOST_R34_10_PUBKEY): идентификатор кривой (curve.id), проективные координаты точки на эллиптической кривой (point.x, point.y, point.z), модуль кривой (curve.dwModLen).
serializeToBufSize	Размер буфера serializeToBuf. Если указан недостаточный размер, то будет возвращена ошибка GOST_E_BUFFER_TOO_SMALL.

2.2.8 Gost_R34_10_2012_deserialize()

Данная функция предназначена для установки параметров открытого ключа в контекст.

```
int Gost_R34_10_2012_deserialize(const void* deserailizeFromBuf, size_t deserailizeFromBufSize,
void* contextBuffer, size_t* contextBufferSize, GOST_R34_10_HANDLE* outHandle)
```

deserailizeFromBuf	Буфер с данными для установки в контекст. Если передан нулевой указатель, то будет возвращен код ошибки GOST_E_INVALID_PARAM. При разборе данных из буфера проверяется корректность указанных значений. В начале буфера указывается идентификатор кривой. Если идентификатор не соответствует ни одной из кривых, указанных в разделе «Экспериментальная проверка корректности», то возвращается код ошибки GOST_E_INVALID_PARAM. Из буфера копируются значения координат точки эллиптической кривой (x, y, z) в контекст contextBuffer и проверяется принадлежность этой точки указанной кривой. В случае ошибки возвращается код GOST_E_INVALID_PARAM.
deserailizeFromBufSize	Размер буфера deserailizeFromBuf. Осуществляется проверка данного значения. Если значение меньше минимальной длины, то возвращается ошибка GOST_E_INVALID_PARAM.
contextBuffer	Указатель на структуру GOST_R34_10_PUBKEY, в которую записываются данные из deserailizeFromBuf.
contextBufferSize	Размер буфера contextBuffer. Если указан недостаточный размер, то будет возвращена ошибка GOST_E_BUFFER_TOO_SMALL.
outHandle	Выходной дескриптор контекста открытого ключа. Дескриптор соответствует структуре contextBuffer, сформированной на основе переданных данных.

2.2.9 Gost_R34_10_2012_clear_public_key()

Данная функция предназначена для обнуления точки эллиптической кривой в контексте открытого ключа.

```
int Gost_R34_10_2012_clear_public_key(GOST_R34_10_HANDLE handle)
```

handle	Дескриптор контекста открытого ключа. Если дескриптор не передан, то будет возвращен код ошибки GOST_E_INVALID_PARAM. Точка кривой из контекста хэширования (поле point структуры GOST_R34_10_PUBKEY) инициализируется нулями.