ОКБ ОСОБОЕ КОНСТРУКТОРСКОЕ БЮРО
СИСТЕМ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ

УТВЕРЖДЕН

37222406.26.20.40.140.083:08 32–ЛУ

Специальное программное обеспечение «Резидентный компонент безопасности»

Руководство системного программиста

37222406.26.20.40.140.083:08 32

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ

ICSP	– In Circuit Serial Programming
USB	– Universal Serial Bus
ПО	– Программное обеспечение
РКБ	– Резидентный компонент безопасности
СДЗ	– Средство доверенной загрузки
СПО	– Специальное программное обеспечение

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ

1.1 Специальное программное обеспечение «Резидентный компонент безопасности», 37222406.26.20.40.140.083:08 (СПО «РКБ») предназначено для выполнения отчужденных от центрального процессора функций безопасности.

1.2 СПО «РКБ» имеет следующие функциональные возможности:

– генерацию, хранение и использование ключевой информации;
– хранение базы данных СПО СДЗ «Аккорд-МКТ»;
– передача случайных чисел от датчика случайных чисел;
– поддержка интерфейса USB.

2 СТРУКТУРА ПРОГРАММЫ

2.1 Текст программы написан на языке С.

2.2 СПО «РКБ» реализовано в виде бинарного файла (файла firmware) и в заводских условиях устанавливается на микроконтроллер РКБ по технологии ICSP.

3 РАБОТА С ПРОГРАММОЙ

3.1 Работа с СПО «РКБ», установленным на микроконтроллер РКБ, производится из прикладного ПО через прикладной интерфейс, реализуемый библиотекой librkb.

3.2 Набор функций, предоставляемых библиотекой:

- генерация неизвлекаемого ключа подписи;
- получение ключа проверки подписи;
- генерация сессионного ключа;
- расшифрование сессионного ключа;
- получение последовательности случайных чисел;
- установка ключа подписи;
- выработка подписи;
- сброс ключа подписи;
- проверка подписи;
- установка сертификата;
- получение сертификата;
- получение информации об устройстве.

4 ФУНКЦИИ БИБЛИОТЕК

4.1 Генерация неизвлекаемого ключа подписи

Команда подается прикладным ПО. По этой команде на основе случайной последовательности байт, полученных с ФДСЧ, firmware РКБ создает неизвлекаемый ключ подписи и записывает с во внутреннюю память РКБ. В РКБ одновременно может хранится только один неизвлекаемый ключ. Соответственно, при генерации нового неизвлекаемого ключа старый ключ стирается. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1] .

Название команды: *rkb_cmd_gen_sign_key*.

Входные параметры: отсутствуют.

Выходные данные: код ошибки с возможными значениями: *SIGN_KEY_GEN_ERR, SUCCESS* (таблица 1).

4.2 Получение ключа проверки подписи

Команда подается прикладным ПО. По этой команде вырабатывается ключ проверки подписи, соответствующий неизвлекаемому ключу подписи, и возвращается в выходных данных. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1].

Название команды: *rkb_cmd_gen_verify_key*.

Входные параметры: отсутствуют.

Выходные данные: код ошибки с возможными значениями: *VERIFY_KEY_BUFFER_TOO_SMALL, VERIFY_KEY_GEN_ERR, SUCCESS* (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
verify_key_len	Длина ключа проверки подписи	4 байта
verify_key	Открытый ключ проверки подписи (порядок байт little-endian)	128 байт

4.3 Генерация сессионного ключа

Команда подается прикладным ПО. По этой команде вырабатывается сессионный ключ, соответствующий открытому ключу, и передается в прикладное ПО, вызвавшее функцию. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1].

Название команды: *rkb_cmd_gen_session_key*.

Входные параметры: передаваемые данные.

[1] Только в firmware СПО «РКБ» версии 23. Номер версии СПО «РКБ» можно узнать в значении поля fw_version (последние 2 байта) при выполнении команды «Получение информации об устройстве» (п. 4.12).

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
verify_key_len	Длина массива ключа проверки подписи	4 байта
verify_key	Открытый ключ проверки подписи	128 байт

Выходные данные: код ошибки с возможными значениями: *VERIFY_KEY_BUFFER_TOO_SMALL, SESSION_KEY_BUFFER_TOO_SMALL, CRYPTED_SESSION_KEY_BUFFER_TOO_SMALL, SESSION_KEY_GEN_ERR, SUCCESS* (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
session_key_len	Длина сессионного ключа	4 байта
session_key	Сессионный ключ (в открытом виде)	32 байта
crypted_session_key_len	Длина зашифрованного сессионного ключа	4 байта
crypted_session_key	UKM и сессионный ключ (конкатенация значений), зашифрованный на ключе, выработанном по протоколу Диффи-Хеллмана на паре ключей из сохранённого внутри устройства неизвлекаемого ключа подписи и переданного на вход открытого ключа получателя	64 байта

4.4 Расшифрование сессионного ключа

Команда подается прикладным ПО. По этой команде зашифрованный сессионный ключ расшифровывается на ключе защиты ключей, который в свою очередь вырабатывается на основе неизвлекаемого ключа подписи и ключа проверки подписи. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1].

Название команды: *rkb_cmd_extract_session_key*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
verify_key_len	Длина открытого ключа проверки подписи	4 байта
verify_key	Открытый ключ проверки подписи	128 байт
crypted_session_key_len	Длина зашифрованного сессионного ключа	4 байта
crypted_session_key	Сессионный ключ, зашифрованный на ключе, выработанном по протоколу Диффи-Хеллмана на паре ключей из сохранённого внутри устройства неизвлекаемого ключа подписи и переданного на вход открытого ключа получателя	32 байта

Выходные данные: код ошибки с возможными значениями: *VERIFY_KEY_BUFFER_TOO_SMALL,*

CRYPTED_SESSION_KEY_BUFFER_TOO_SMALL, EXTRACT_SESSION_KEY_ERR, SUCCESS (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
session_key_len	Длина сессионного ключа	4 байта
session_key	Сессионный ключ (в открытом виде)	32 байта

4.5 Получение последовательности случайных чисел

Команда подается прикладным ПО. По этой команде происходит получение случайных чисел с ФДСЧ, их обработка, контроль качества и запись в выходной массив случайных чисел.

Название команды: *rkb_cmd_get_rnd*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
len	Длина массива случайных чисел	4 байта

Выходные данные: код ошибки с возможными значениями: *RND_GEN_ERR, SUCCESS* (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
buf	Массив случайных чисел	В соответствии с размером len

4.6 Установка ключа подписи

Команда подается прикладным ПО. По этой команде происходит запись во внутреннюю память РКБ ключа подписи.

Название команды: *rkb_cmd_set_sign_key*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
sign_key_len	Длина закрытого ключа подписи	4 байта
sign_key	Закрытый ключ подписи (порядок байт little-endian)	64 байта

Выходные данные: код ошибки с возможными значениями: *SIGN_KEY_SET_ERROR, SUCCESS* (таблица 1).

4.7 Выработка подписи

Команда подается прикладным ПО. По этой команде происходит выработка подписи на основе ключа подписи и массива со значением хэша. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1].

Название команды: *rkb_cmd_sign*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
hash_len	Длина подписываемых данных	4 байта
hash	Подписываемые данные (хеш-значение) (порядок байт big-endian[1])	64 байта

Выходные данные: код ошибки с возможными значениями: *SIGN_ERR, SUCCESS* (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
sign_len	Длина подписи	4 байта
sign	Значение подписи (s|r порядок байт little-endian)	128 байт

4.8 Сброс ключа подписи

Команда подается прикладным ПО. По этой команде происходит удаление ключа подписи из внутренней памяти РКБ.

Название команды: *rkb_cmd_reset*.

Входные параметры: отсутствуют.

Выходные данные: код ошибки с возможными значениями: *RESET_ERR, SUCCESS* (таблица 1).

4.9 Проверка подписи

Команда подается прикладным ПО. По этой команде происходит проверка подписи на базе ключа подписи, массива со значением хэша и массива со значением подписи. При вычислении используется набор параметров *id-tc26-gost-3410-12-512-paramSetA*[1].

Название команды: *rkb_cmd_verify*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
hash_len	Длина подписываемых данных	4 байта
hash	Подписываемые данные (хеш-значение) (порядок байт big-endian[1])	64 байта
sign_len	Длина подписи	4 байта
Sign	Значение подписи (s|r порядок байт little-endian)	128 байт

Выходные данные: код ошибки с возможными значениями: *HASH_BUFFER_TOO_SMALL, SIGN_BUFFER_TOO_SMALL, VERIFY_ERR, SUCCESS* (таблица 1).

4.10 Установка сертификата

Команда подается прикладным ПО. По этой команде происходит запись во внутреннюю память РКБ сертификата. В энергонезависимой памяти устройства выделяется блок размером 8Кб для хранения сертификата. В случае передачи для сохранения сертификата большего размера произойдёт ошибка записи. Проверка данных сертификата устройством не выполняется.

Название команды: *rkb_cmd_set_cert*.

Входные параметры: передаваемые данные.

Формат передаваемых данных:

Имя данных	Назначение	Размер данных
cert_len	Длина массива с сертификатом	4 байта
cert	Массив с сертификатом	В соответствии со значением cert_len

Выходные данные: код ошибки с возможными значениями: *TOO_MUCH_DATA, SET_CERT_ERR, SUCCESS* (таблица 1).

4.11 Получение сертификата

Команда подается прикладным ПО. По этой команде происходит чтение сертификата из внутренней памяти РКБ.

Название команды: *rkb_cmd_get_cert*.

Входные параметры: отсутствуют.

Выходные данные: код ошибки с возможными значениями: *GET_CERT_ERR, SET_CERT_ERR, SUCCESS* (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
cert_len	Длина массива с сертификатом	4 байта
cert	Массив с сертификатом	В соответствии со значением cert_len

4.12 Получение информации об устройстве

Команда подается прикладным ПО. По этой команде происходит получение информации о серийном номере и версии СПО «РКБ».

Название команды: *rkb_cmd_get_info*.

Входные параметры: отсутствуют.

Выходные данные: код ошибки с возможными значениями: GET_INFO_ERR, SUCCESS (таблица 1), возвращаемые данные.

Формат возвращаемых данных:

Имя данных	Назначение	Размер данных
serial_number	Серийный (заводской) номер (пример: «12345678»)	9 байт
fw_version	Версия прошивки (firmware) СПО «РКБ» (пример: «1.0.0023»)	14 байт

5 СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ

5.1 Перечень сообщений, получаемых системным программистом при работе с СПО «РКБ», отражен в таблице 1.

Таблица 1 – Перечень сообщений от СПО «РКБ»

Сообщение	Код
#define SUCCESS	0x0
#define VERIFY_KEY_BUFFER_TOO_SMALL	0x201
#define SESSION_KEY_BUFFER_TOO_SMALL	0x202
#define CRYPTED_SESSION_KEY_BUFFER_TOO_SMALL	0x203
#define SIGN_KEY_GEN_ERR	0x204
#define VERIFY_KEY_GEN_ERR	0x205
#define SESSION_KEY_GEN_ERR	0x206
#define EXTRACT_SESSION_KEY_ERR	0x207
#define RND_GEN_ERR	0x208
#define SIGN_KEY_BUFFER_TOO_SMALL	0x209
#define SIGN_KEY_SET_ERROR	0x20A
#define HASH_BUFFER_TOO_SMALL	0x20B
#define SIGN_ERR	0x20C
#define RESET_ERR	0x20D
#define SIGN_BUFFER_TOO_SMALL	0x20E
#define VERIFY_ERR	0x20F
#define TOO_MUCH_DATA	0x210
#define SET_CERT_ERR	0x211
#define GET_CERT_ERR	0x212
#define GET_INFO_ERR	0x213

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номер изме-нения	Номер листа				Номер и дата документа	Дата внесения измен-ения	Подпись лица, внесшего изменения
	измененн ого	замененно го	нового	анулиро-ванного			

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм	Лист	№ докум.	Подп.	Дата	37222406.26.20.40.140.083:08 32	Лист 13