

***Платформа удалённого обновления  
Senselock Remote Update (RU)  
(принцип реализации, описание, API).***



## Содержание.

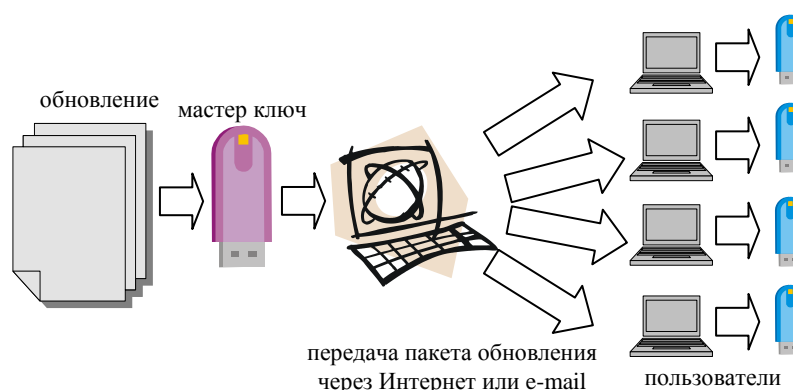
Содержание.....	2
1. Вступление.....	3
1.1. Структурная схема системы обновления.....	3
1.2. Клиентская часть системы удалённого обновления.....	4
1.3. Мастер ключ системы удалённого обновления.....	4
1.4. Структурная схема пакета обновления.....	5
2. Описание функций API “E4RUInit.dll”.....	6
2.1. e4ru_CreateClientDevice.....	6
2.2. e4ru_CreateIssuerDevice.....	7
2.3. e4ru_ReleaseFile.....	8
2.4. Возвращаемые значения функциями E4RU_Init.DLL.....	9
3. Описание функций API “E4RUIssuer.dll”.....	10
3.1 e4ru_AddFile.....	10
3.2 e4ru_AddModule.....	11
3.3 e4ru_InitModule.....	12
3.4 e4ru_InitPkg.....	13
3.5 e4ru_ProducePkg.....	15
3.6 e4ru_ReleaseModule.....	16
3.7 e4ru_ReleasePkg.....	17
3.8 Возвращаемые значения функциями E4RU_Issuer.DLL.....	18
3.9 Типы файлов.....	18
3.10 Флаги действия при обновлении.....	19
4 Описание функций API “E4RUClient.dll”.....	20
4.1 e4ru_GetDeviceInfo.....	20
4.2 e4ru_GetDeviceModuleInfo.....	21
4.3 e4ru_GetPkgInfo.....	22
4.4 e4ru_GetPkgModuleInfo.....	23
4.5 e4ru_UpdatePkg.....	24
4.6 Возвращаемые значения функциями E4RU_Client.DLL.....	25
4.7 Описание структуры SYSTEM_INFORMATION.....	26
4.8 Описание структуры DEVICE_MODULE_INFO.....	27
4.9 Описание структуры PACKAGE_INFORMATION.....	28
4.10 Описание структуры PACKAGE_MODULE_INFO.....	29
Где купить наши продукты.....	30
История документа.....	31

## 1. Вступление.



Система удалённого обновления (RU – Remote Update) ключа предназначена для организации безопасной схемы управления контентом системы защиты. В частности изменения внутренних файлов данных, исполняемых модулей и т.д. удалённо, посредством передачи зашифрованного файла через Интернет, e-mail или включая пакеты обновления в новые версии защищенного программного обеспечения.

### 1.1. Структурная схема системы обновления.



На диаграмме представлена процедура удалённого обновления ключей.

- Первый этап обновления, это подготовка необходимых файлов, которые в последствие будут обновлены в ключах пользователей.
- Далее следует создание защищенного пакета обновления с помощью мастер ключа и отправка его пользователям.
- Третий этап, это получение клиентом защищенного пакета обновления и его установка.

Система удаленного обновления состоит из трёх основных модулей:

- 1) **INIT** - предназначен для создания защищенной пары мастер ключа и клиентского модуля. Возможности API позволяют производить подготовку и программирование клиентских и мастер ключей, а также имеется возможность получить сгенерированные BIN и DAT файлы мастер модуля и клиентского модуля.
- 2) **ISSUER** – предназначен для создания пакета обновления и установки необходимых параметров, таких как назначение обновление определённым ключам, ограничение количества обновлений...
- 3) **CLIENT** – предназначен для установки пакета обновления в клиентский ключ. Также API этого модуля позволяет получать информацию из ключа об уже установленных обновлениях и ID ключа.

## 1.2. Клиентская часть системы удалённого обновления.



Клиентская часть системы обновления осуществляет перезапись внутренних файлов данных и исполняемых модулей внутри ключа. Она осуществляет контроль целостности пакета обновления, версию, ID устройства которому оно предназначено.

**! Внимание! Клиентскую часть системы обновлений должна быть предусмотрена и установлена в клиентские ключи на этапе проектирования защиты!**



Клиентское приложение для ключа состоит из двух модулей:

1. - Основной исполняемый модуль с ID: **FB11**
2. - Информационный модуль данных с ID: **FB12**

Основной исполняемый модуль содержит в себе необходимые для передачи обновления в ключ коммуникационные решения и механизмы контроля целостности.

Информационный модуль содержит информацию о произведённых обновлениях, количестве установленных модулей и необходимую внутреннюю техническую информацию.

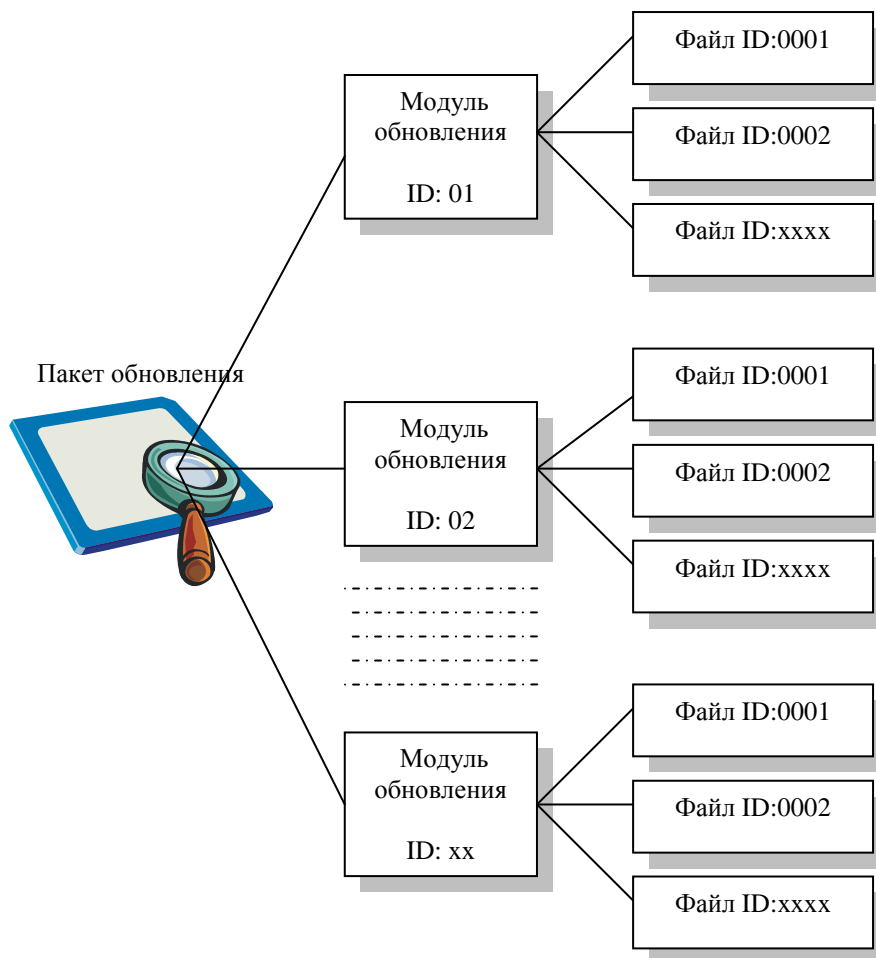
## 1.3. Мастер ключ системы удалённого обновления.



Мастер ключ и клиентский модуль имеют одну уникальную пару ключей шифрования для надёжной защиты и контроля целостности пакетов обновления. Мастер ключ необходим для обеспечения безопасности и предотвращения неконтролируемого распространения секретного ключа, с помощью которого производится подготовка и шифрование пакетов обновлений. Производитель программного обеспечения может создать любое необходимое ему количество мастер ключей и распределить их среди программистов, которые имеют доступ к созданию и обновлению программ и данных ключа.

Приложение мастер ключа состоит из одного модуля с ID: **FB01** с сохранённым в нем секретным ключом для шифрования пакетов обновлений.

#### 1.4. Структурная схема пакета обновления.



На приведенной выше схеме представлена структура пакета обновления. Пакет обновления состоит из модулей. Модули в свою очередь содержат файлы, которые будут обновлены. Пакет обновления имеет ряд параметров направленных на возможность провести обновление, как конкретного ключа, так и целой серии ключей. Предусмотрена возможность указания, как однократного обновления, так и многократного. Каждому из модулей присваивается уникальное ID а также название и версия. В модуль обновления можно включить не более 255 файлов. При формировании модуля обновления пользователь может добавлять файлы с дополнительными параметрами и атрибутами. Например, файлы данных (к исполняемым файлам это не относится!) можно обновлять только частично, указав смещение с которого надо произвести обновление. Также при добавлении файла есть возможность зарезервировать дополнительное свободное пространство под файл, указав планируемый размер файла.

## 2. Описание функций API “E4RUInit.dll”

### 2.1. e4ru\_CreateClientDevice

Функция подготавливает клиентский ключ.

```
DWORD WINAPI e4ru_CreateClientDevice (  
    IN    PSENSE4_CONTEXT    pS4Ctx,  
    IN    BYTE                *pDevPin,  
    IN    BYTE                *pTDESA  
);
```

#### Параметры:

- |                |   |
|----------------|---|
| <i>pS4Ctx</i>  | [in] Указатель на структуру подключенного ключа.  |
| <i>pDevPin</i> | [in] Указатель на PIN код разработчика (24 символа), если указывает на NULL то указатель автоматически будет установлен на PIN код по умолчанию “123456781234567812345678”. |
| <i>pTDESA</i>  | [in] Указатель на секретный ключ для шифрования пакетов обновлений. Ключ длиной 16 байт. Необходимо держать в секрете!  |

#### Возвращаемый результат:

Если функция завершила работу без ошибок, будет возвращен результат E4RU\_SUCCESS.

Расшифровку других возможных ошибок смотрите в [Параграфе 2.4 Таблица 1.](#)

#### Комментарий:

-

#### Требования:

Senselock EL с микропрограммой версии 2.3.0 и выше.

## 2.2. e4ru\_CreateIssuerDevice

Функция подготавливает мастер ключ

```

DWORD WINAPI e4ru_CreateIssuerDevice (
    IN    PSENSE4_CONTEXT    pS4Ctx,
    IN    LPCSTR              lpDirectory,
    IN    DWORD               Size,
    IN    DWORD               dwEraseFlag,
    IN    BYTE                *pDevPin,
    IN    BYTE                *pTDESA,
    IN    BYTE                ModuleNum,
    IN    BYTE                ErrorMax
);

```

### Параметры:

<i>pS4Ctx</i>	[in]	Указатель на структуру подключенного ключа.
<i>lpDirectory</i>	[in]	ID каталога
<i>Size</i>	[in]	Размер каталога. Рекомендуемое значение более 5120 байт.
<i>dwEraseFlag</i>	[in]	Режим работы системы обновления. Смотри комментарий.
<i>pDevPin</i>	[in]	Указатель на PIN код разработчика (24 символа)
<i>pTDESA</i>	[in]	Указатель на секретный ключ для шифрования пакетов обновлений. Ключ длиной 16 байт. Необходимо держать в секрете!
<i>ModuleNum</i>	[in]	Максимальное количество модулей в пакете обновления. Значение от 1 до 255.
<i>ErrorMax</i>	[in]	Максимально возможное количество ошибок при обновлении. Значение от 1 до 255.

### Возвращаемый результат:

Если функция завершила работу без ошибок, будет возвращен результат E4RU\_SUCCESS.

Расшифровку других возможных ошибок смотрите в [Параграфе 2.4 Таблица 1.](#)

### Комментарий:

Режим работы клиентской части системы обновлений:

E4RU_ERASE_DIR	–	Перед обновлением каталог с указанным ID будет сперва удалён вместе со всем содержимым
E4RU_UPDATE_FILE	–	Обновляться будут только файлы, которые содержатся в пакете обновлений.

### Требования:

Senselock EL с микропрограммой версии 2.3.2 и выше.

### 2.3. e4ru\_ReleaseFile

Функция создаёт исполняемые файлы мастер ключа и клиентского ключа.

```

DWORD WINAPI e4ru_ReleaseFile (
    IN         BYTE      *pTDESA,
    IN         BYTE      ModuleNum,
    IN         BYTE      ErrorMax,
    IN/OUT     DWORD     *pdwIssuerLen,
    IN/OUT     DWORD     *pdwClientLen,
    IN/OUT     DWORD     *pdwInfoLen,
    OUT        BYTE      *pIssuerContent,
    OUT        BYTE      *pClientContent,
    OUT        BYTE      *pInfoContent
);

```

#### Параметры:

<i>pTDESA</i>	[in]	Указатель на секретный ключ для шифрования пакетов обновлений. Ключ длиной 16 байт. Необходимо держать в секрете!
<i>ModuleNum</i>	[in]	Максимальное количество модулей в пакете обновления. Значение от 1 до 255.
<i>ErrorMax</i>	[in]	Максимально возможное количество ошибок при обновлении. Значение от 1 до 255.
<i>pdwIssuerLen</i>	[in][out]	Размер исполняемого модуля мастер ключа.
<i>pdwClientLen</i>	[in][out]	Размер исполняемого модуля клиентского ключа.
<i>pdwInfoLen</i>	[in][out]	Размер информационного файла.
<i>pIssuerContent</i>	[out]	Содержимое исполняемого модуля мастер ключа.
<i>pClientContent</i>	[out]	Содержимое клиентского модуля ключа.
<i>pInfoContent</i>	[out]	Содержимое информационного файла.

#### Возвращаемый результат:

Если функция завершила работу без ошибок, будет возвращен результат E4RU\_SUCCESS.

Расшифровку других возможных ошибок смотрите в [Параграфе 2.4 Таблица 1.](#)

#### Комментарий:

-

#### Требования:

-



**2.4. Возвращаемые значения функциями E4RU\_Init.DLL**

Таблица 1

Код ошибки	Обозначение	Комментарий
0x00000000	<b>E4RU_SUCCESS</b>	ОК
0xE0002002	<b>E4RU_INVALID_PARAMETER</b>	Неверный параметр
0xE0002003	<b>E4RU_INSUFFICIENT_BUFFER</b>	Нехватка буфера
0xE0002004	<b>E4RU_DEVICE_NOT_FOUND</b>	Ключ не найден
0xE0002101	<b>E4RU_INVALID_DEVICE_VERSION</b>	Версия микропрограммы ключа ниже требуемой
0xE0002102	<b>E4RU_OPEN_DEVICE_ERROR</b>	Ошибка доступа к ключу
0xE0002103	<b>E4RU_INSUFFICIENT_MEMORY</b>	Нехватка памяти
0xE0002104	<b>E4RU_CREATE_DIR_ERROR</b>	Ошибка создания директория
0xE0002105	<b>E4RU_ERASE_DIR_ERROR</b>	Ошибка удаления директория
0xE0002106	<b>E4RU_VERIFYPIN_ERROR</b>	Ошибка проверки PIN кода
0xE0002107	<b>E4RU_WRITEFILE_ERROR</b>	Ошибка записи файла
0xE0002108	<b>E4RU_INSUFFICIENT_DIR</b>	Не найдена директория
0xE0002109	<b>E4RU_INVALID_DIRNAME</b>	Неверное имя директории
0xE000210A	<b>E4RU_UPDATE_FILE_OVERSIZE</b>	Неверный размер обновляемого файла
0xE000210B	<b>E4RU_GET_FILE_INFO_ERROR</b>	Ошибка получения информации о файле
0xE000210C	<b>E4RU_MISMATCH_FILE_TYPE</b>	Неверный тип файла
0xE000210D	<b>E4RU_GET_DEVICE_TYPE_ERROR</b>	Ошибка определения типа ключа
0xE000210E	<b>E4RU_MISMATCH_DEVICE_TYPE</b>	Неверный тип ключа
0xE000210F	<b>E4RU_GET_AVAILABLE_SPACE_ERROR</b>	Ошибка получения доступного размера памяти ключа

### 3. Описание функций API “E4RUIssuer.dll”

#### 3.1 e4ru\_AddFile

Добавить файл данных или исполняемый файл в модуль пакета обновления.

```

DWORD WINAPI e4ru_AddFile(
    IN HANDLE ModuleHandle,
    IN WORD   EfFileID,
    IN BYTE   FileType,
    IN WORD   Size,
    IN WORD   Offset,
    IN BYTE   *Content,
    IN WORD   ContentLen
);

```

#### Параметры:

<i>ModuleHandle</i>	[in] Указатель на структуру (HANDLE), полученную после применения функций <code>e4ru_InitModule</code> или <code>e4ru_CreateModule</code> .
<i>EfFileID</i>	[in] ID добавляемого в пакет обновления файла. Не может быть равным “FF01”.
<i>FileType</i>	[in] Тип добавляемого файла. Возможные значения перечислены в <a href="#">Параграфе 3.9 Таблица 2</a> . Если указано иное значение – добавление не будет успешным.
<i>Size</i>	[in] Размер добавляемого файла. Не может быть равным 0.
<i>Offset</i>	[in] Начальный адрес обновляемых данных. Используется в случае если необходимо обновить только часть файла. Если требуется обновить весь файл, то значение должно быть равным 0.
<i>Content</i>	[in] Указатель на данные добавляемого файла.
<i>ContentLen</i>	[in] Длина добавляемых данных.

#### Возвращаемый результат:

Если добавление файла прошло успешно, возвращается код `E4RU_SUCCESS`. В случае ошибки смотрите коды в [Параграфе 3.8 Таблица 1](#).

#### Комментарий:

<i>ModuleHandle</i>	– не должен быть равен 0 (NULL).
<i>Size</i>	– может быть больше чем значения <i>ContentLen</i> и <i>Offset</i>
<i>FileType</i>	– исполняемые модули должны быть только в BIN формате. Если добавляемый файл является исполняемым модулем и <i>Offset</i> указывает не на 0, то файл не будет добавлен.
<i>EfFileID</i>	– если ID добавляемого файла уже содержится в пакете обновления, новый файл добавляться не будет. В пакете обновлений может быть не более 255 файлов.

#### Требования:

-

### 3.2 e4ru\_AddModule

Добавить модуль в пакет обновления.

```
DWORD WINAPI e4ru_AddModule(  
    IN HANDLE PkgHandle,  
    IN HANDLE ModuleHandle  
);
```

#### Параметры:

<i>PkgHandle</i>	[in] указатель на структуру, полученную после использования функции e4ru_InitPkg
<i>ModuleHandle</i>	[in] указатель на структуру, полученную после использования функции e4ru_CreateModule

#### Возвращаемый результат:

Если добавление файла прошло успешно, возвращается код E4RU\_SUCCESS.  
В случае ошибки смотрите коды в [Параграфе 3.8 Таблица 1](#).

#### Комментарий:

-

#### Требования:

-

### 3.3 e4ru\_InitModule

Инициализация создания модуля обновления.

```
HANDLE WINAPI e4ru_InitModule(  
    IN  BYTE    Mid,  
    IN  DWORD   Version,  
    IN  LPCSTR  lpName  
);
```

**Параметры:**

*Mid* [in] ID модуля обновления. Должен быть больше 0.  
*Version* [in] Версия модуля обновления.  
*lpName* [in] Название модуля обновления. Длина не более 16 символов.

**Возвращаемый результат:**

Если инициализация прошла успешно, возвращается указатель на структуру (HANDLE). В противном случае возвращается 0.

**Комментарий:**

Если имя модуля обновления содержит более 16 символов, производится автоматическая коррекция строки до 16 символов.

**Требования:**

-

### 3.4 e4ru\_InitPkg

Инициализирует пакет обновления и назначает ему необходимые параметры

```
HANDLE WINAPI e4ru_InitPkg(
    IN  BYTE    Scope,
    IN  BYTE    ValidType,
    IN  BYTE    UpdateType,
    IN  BYTE    *pDeviceID,
    IN  DWORD   SequenceNum
);
```

#### Параметры:

- Scope* [in] Флаг использования пакета обновления.  
Смотрите в [Параграфе 3.10 Таблица 3](#).  
Возможные значения:
- E4RU\_FLAG\_ALL\_USER (можно использовать на любых ключах)
  - E4RU\_FLAG\_INDIVIDUAL\_USER (предназначен определённому ключу, серийный номер которого указан в параметре *pDeviceID*)
- ValidType* [in] Флаг действия обновления.  
Смотрите в [Параграфе 3.10 Таблица 3](#).  
Возможные значения:
- E4RU\_FLAG\_UPDATE\_ONCE (действует единожды)
  - E4RU\_FLAG\_UPDATE\_PERMANENT (действует неограниченное количество раз)
- UpdateType* [in] Флаг типа обновления.  
Смотрите в [Параграфе 3.10 Таблица 3](#).  
Возможные значения:
- E4RU\_FLAG\_CREATE\_NEW\_MODULE (создавать новые модули)
  - E4RU\_FLAG\_UPDATE\_EXISTING\_ONLY (обновлять только установленные модули)
- pDeviceID* [in] Указатель на 8 байт уникального серийного номера ключа, которому предназначен пакет обновления. Необходим только в случае использования опции E4RU\_FLAG\_INDIVIDUAL\_USER
- SequenceNum* [in] Номер обновления. Может использоваться только при выставленном флаге E4RU\_FLAG\_UPDATE\_ONCE параметра *ValidType*. Должен иметь значение, на единицу больше количества установленных обновлений в ключе-клиенте. Количество установленных обновлений можно получить посредством вызова функции `e4ru_GetDeviceInfo()`, оно будет содержаться в структуре `SYSTEM_INFORMATION`, в параметре *GlobalSerial* (если установлен флаг E4RU\_FLAG\_ALL\_USER) или *PrivateSerial* (если установлен флаг E4RU\_FLAG\_INDIVIDUAL\_USER).

#### Возвращаемый результат:

Если инициализация прошла успешно, возвращается указатель на структуру (HANDLE). В противном случае возвращается 0.

**Комментарий:**

-

**Требования:**

-

### 3.5 e4ru\_ProducePkg

Шифрует созданный пакет обновления с помощью мастер ключа.

```
DWORD WINAPI e4ru_ProducePkg(  
    IN         PSENSE4_CONTEXT    pS4Ctx,  
    IN         BYTE                *UserPin,  
    IN         HANDLE              PkgHandle,  
    OUT        BYTE                *Buffer,  
    IN OUT    DWORD                *BufferLen  
);
```

#### Параметры:

*pS4Ctx* [IN] Указатель на структуру подключенного ключа.  
*UserPin* [IN] PIN код пользователя.  
*PkgHandle* [IN] Указатель на структуру пакета сформированного пакета обновления.  
*Buffer* [OUT] Буфер, содержащий полученный пакет обновления.  
*BufferLen* [IN][OUT] Длина полученного буфера.

#### Возвращаемый результат:

Если добавление файла прошло успешно, возвращается код E4RU\_SUCCESS.  
В случае ошибки смотрите коды в [Параграфе 3.8 Таблица 1](#).

#### Комментарий:

-

#### Требования:

-

### 3.6 e4ru\_ReleaseModule

Освобождает память, зарезервированную модулем пакета обновления.

```
DWORD WINAPI e4ru_ReleaseModule(  
    IN HANDLE ModuleHandle  
);
```

**Параметры:**

*ModuleHandle* [in] Указатель на структуру (HANDLE) модуля созданного с помощью функции e4ru\_CreateModule.

**Возвращаемый результат:**

Если добавление файла прошло успешно, возвращается код E4RU\_SUCCESS.  
В случае ошибки смотрите коды в [Параграфе 3.8 Таблица 1](#).

**Комментарий:**

-

**Требования:**

-



### 3.7 e4ru\_ReleasePkg

Освобождает память, зарезервированную пакетом обновлений.

```
DWORD WINAPI e4ru_ReleasePkg(  
    IN HANDLE PkgHandle  
);
```

**Параметры:**

*PkgHandle* [in] Указатель на структуру (HANDLE) пакета обновлений созданного с помощью функции `e4ru_InitPkg`.

**Возвращаемый результат:**

Если добавление файла прошло успешно, возвращается код `E4RU_SUCCESS`.  
В случае ошибки смотрите коды в [Параграфе 3.8 Таблица 1](#).

**Комментарий:**

-

**Требования:**

-

**3.8 Возвращаемые значения функциями E4RU\_Issuer.DLL**

Таблица 1

Код ошибки	Обозначение	Комментарий
0x00000000	<b>E4RU_SUCCESS</b>	ОК
0xE0002001	<b>E4RU_ERROR</b>	Ошибка выполнения функции
0xE0002002	<b>E4RU_INVALID_PARAMETER</b>	Неверный параметр
0xE0002003	<b>E4RU_INSUFFICIENT_BUFFER</b>	Нехватка буфера
0xE0002004	<b>E4RU_DEVICE_NOT_FOUND</b>	Ключ не найден
0xE0002005	<b>E4RU_FILE_TYPE_MISMATCH</b>	Неправильный тип файла
0xE0002006	<b>E4RU_MODULE_NO_CONTENT</b>	Модуль не содержит данные
0xE0002007	<b>E4RU_PACKAGE_NO_CONTENT</b>	Пакет обновления не содержит данные
0xE0002008	<b>E4RU_FILE_CREATE_SIZE_ERROR</b>	Размер создаваемого файла неверный
0xE0002009	<b>E4RU_FILE_NAME_RESERVED</b>	Имя файла зарезервировано
0xE0002201	<b>E4RU_INVALID_HANDLE</b>	Указатель (HANDLE) не правильный
0xE0002203	<b>E4RU_GEN_BLOCK_FAIL</b>	Ошибка генерации блока
0xE0002204	<b>E4RU_DIGEST_ERROR</b>	Ошибка подписи
0xE0002205	<b>E4RU_SPACE_NOT_ENOUGH</b>	Недостаточно свободного места
0xE0002206	<b>E4RU_NOT_ISSUER_DEVICE</b>	Нет мастер ключа
0xE0002207	<b>E4RU_OFFSET_ERROR</b>	Ошибка в смещении
0xE0002208	<b>E4RU_SAME_EF_ID_EXIST</b>	Такой ID уже существует
0xE0002209	<b>E4RU_SAME_MODULE_ID_EXIST</b>	Такой модуль уже существует
0xE0002210	<b>E4RU_EF_FILE_OVERNUMBER</b>	Количество возможных файлов превышено

**3.9 Типы файлов**

Таблица 2

Код	Обозначение	Комментарий
0x00	<b>E4RU_FILE_TYPE_EXE</b>	Исполняемый модуль
0x01	<b>E4RU_FILE_TYPE_EXE_DATA</b>	Модуль данных
0x02	<b>E4RU_FILE_TYPE_RSA_PUB</b>	Публичный RSA ключ
0x03	<b>E4RU_FILE_TYPE_RSA_SEC</b>	Приватный RSA ключ

### 3.10 Флаги действия при обновлении

Таблица 3

Код	Обозначение	Комментарий
0x01	<b>E4RU_FLAG_CREATE_NEW_MODULE</b>	Создать новый модуль
0x02	<b>E4RU_FLAG_UPDATE_EXISTING_ONLY</b>	Обновить только существующий модуль
0x04	<b>E4RU_FLAG_UPDATE_ONCE</b>	Обновить файл только единожды
0x08	<b>E4RU_FLAG_UPDATE_PERMANENT</b>	Разрешить обновлять файл неограниченно
0x10	<b>E4RU_FLAG_ALL_USER</b>	Разрешить обновлять всем пользователям
0x12	<b>E4RU_FLAG_INDIVIDUAL_USER</b>	Обновление строго индивидуально

## 4 Описание функций API “E4RUClient.dll”

### 4.1 e4ru\_GetDeviceInfo

Получает информацию об установленном в ключ пакете обновления.

```

DWORD WINAPI e4ru_GetDeviceInfo (
    IN     SENSE4_CONTEXT      *pS4Ctx,
    IN     BYTE                *pUserPin,
    IN     LPCSTR              lpDirectory,
    OUT    SYSTEM_INFORMATION  *pInfo,
    OUT    DWORD               *pModuleCount
);

```

#### Параметры:

<i>pS4Ctx</i>	[in] Указатель на структуру подключенного ключа.
<i>pUserPin</i>	[in] Указатель на пользовательский PIN доступа к каталогу содержащему клиентский модуль обновления ключа.
<i>lpDirectory</i>	[in] ID каталога, содержащего клиентский модуль обновления ключа.
<i>pInfo</i>	[out] Указатель на структуру, содержащую полученную информацию о пакете обновления.
<i>pModuleCount</i>	[out] Количество установленных модулей обновлений в ключе

#### Возвращаемый результат:

Если функция выполнена успешно, возвращается код E4RU\_SUCCESS и необходимые данные можно получить из структуры, на которую указывает *pInfo*. В случае ошибки смотрите коды в [Параграфе 4.6 Таблица 1](#). Структура *pInfo* подробно описана в [Параграфе 4.7 Таблица 2](#). Дополнительный возвращаемый параметр *pModuleCount* указывает на количество установленных модулей в ключе.

#### Комментарий:

Если *pS4Ctx* равен NULL, то будет найден первый подключенный к системе ключ. Также при установленном в NULL параметре *pS4Ctx*, библиотека E4RUClient.dll произведёт открытие и закрытие хэнгла ключа самостоятельно внутри себя.

#### Требования:

Senselock EL с микропрограммой версии 2.3.2 и выше.

## 4.2 e4ru\_GetDeviceModuleInfo

Получает информацию о модулях обновлений установленных в ключе.

```

DWORD WINAPI e4ru_GetDeviceModuleInfo (
    IN  SENSE4_CONTEXT      *pS4Ctx,
    IN  BYTE                 *pUserPin,
    IN  LPCSTR               lpDirectory,
    OUT DEVICE_MODULE_INFO  *pModuleInfo,
    IN  DWORD                ModuleInfoLen
);

```

### Параметры:

<i>pS4Ctx</i>	[in] Указатель на структуру подключенного ключа.
<i>pUserPin</i>	[in] Указатель на пользовательский PIN доступа к каталогу содержащему клиентский модуль обновления ключа.
<i>lpDirectory</i>	[in] ID каталога, содержащего клиентский модуль обновления ключа.
<i>pModuleInfo</i>	[out] Указатель на структуру, содержащую полученную информацию о модуле обновления.
<i>ModuleInfoLen</i>	[in] Размер буфера, в который будет помещена структура с информацией о модулях обновлений. Равен количеству модулей помноженному на размер структуры, содержащей информацию о модуле.

### Возвращаемый результат:

Если функция выполнена успешно возвращается код E4RU\_SUCCESS и необходимые данные можно получить из структуры, на которую указывает *pModuleInfo*.

В случае ошибки смотрите коды в [Параграфе 4.6 Таблица 1](#).

Структура *pModuleInfo* подробно описана в [Параграфе 4.8 Таблица 3](#).

### Комментарий:

-

### Требования:

Senselock EL с микропрограммой версии 2.3.2 и выше.

### 4.3 e4ru\_GetPkgInfo

Получает информацию о пакете обновления

```
DWORD WINAPI e4ru_GetPkgInfo (  
    IN  BYTE          *pPkgContent,  
    IN  DWORD         PkgLen,  
    OUT PACKAGE_INFORMATION *pInfo,  
    OUT DWORD         *pModuleCount  
);
```

#### Параметры:

<i>pPkgContent</i>	[in] Указатель на массив, содержащий файл пакета обновления.
<i>PkgLen</i>	[in] Длина массива, содержащего пакет обновления.
<i>pInfo</i>	[out] Указатель на структуру, содержащую полученную информацию о пакете обновления.
<i>pModuleCount</i>	[out] Количество модулей в пакете обновления.

#### Возвращаемый результат:

Если функция выполнена успешно, возвращается код E4RU\_SUCCESS и необходимые данные можно получить из структуры, на которую указывает *pInfo*.

В случае ошибки смотрите коды в [Параграфе 4.6 Таблица 1](#).

Структура *pInfo* подробно описана в [Параграфе 4.9 Таблица 4](#).

Дополнительный возвращаемый параметр *pModuleCount* указывает на количество модулей в пакете обновления.

#### Комментарий:

-

#### Требования:

-

#### 4.4 e4ru\_GetPkgModuleInfo

Получает информацию о модулях из пакета обновления.

```
DWORD WINAPI e4ru_GetPkgModuleInfo (  
    IN  BYTE          *pPkgContent,  
    IN  DWORD         PkgLen,  
    OUT PACKAGE_MODULE_INFO *pModuleInfo,  
    IN  DWORD         ModuleInfoLen  
);
```

##### Параметры:

<i>pPkgContent</i>	[in] Указатель на массив, содержащий файл пакета обновления.
<i>PkgLen</i>	[in] Длина массива, содержащего пакет обновления.
<i>pModuleInfo</i>	[out] Указатель на структуру, содержащую полученную информацию о модуле обновления.
<i>ModuleInfoLen</i>	[in] Размер буфера, в который будет помещена структура с информацией о модулях обновлений. Равен количеству модулей помноженному на размер структуры, содержащей информацию о модуле.

##### Возвращаемый результат:

Если функция выполнена успешно возвращается код E4RU\_SUCCESS и необходимые данные можно получить из структуры, на которую указывает *pModuleInfo*.

В случае ошибки смотрите коды в [Параграфе 4.6 Таблица 1](#).

Структура *pModuleInfo* подробно описана в [Параграфе 4.10 Таблица 5](#).

##### Комментарий:

-

##### Требования:

-

## 4.5 e4ru\_UpdatePkg

Производит установку пакета обновления в клиентский ключ.

```
DWORD WINAPI e4ru_UpdatePkg (  
    IN    SENSE4_CONTEXT    *pS4Ctx,  
    IN    BYTE               *pUserPin,  
    IN    LPCSTR             lpDirectory,  
    IN    BYTE               *pPkgContent,  
    IN    DWORD              PkgLen  
);
```

### Параметры:

*pS4Ctx* [in] Указатель на структуру подключенного ключа.  
*pUserPin* [in] Указатель на пользовательский PIN доступа к каталогу, содержащему клиентский модуль обновления ключа.  
*lpDirectory* [in] ID каталога, содержащего клиентский модуль обновления ключа.  
*pPkgContent* [in] Указатель на массив, содержащий файл пакета обновления.  
*PkgLen* [in] Длина массива, содержащего пакет обновления.

### Возвращаемый результат:

Если функция выполнена успешно возвращается код E4RU\_SUCCESS.  
В случае ошибки смотрите коды в [Параграфе 4.6 Таблица 1](#).

### Комментарий:

-

### Требования:

-



**4.6 Возвращаемые значения функциями E4RU\_Client.DLL**

Таблица 1

Код ошибки	Обозначение	Комментарий
0x00000000	<b>E4RU_SUCCESS</b>	ОК
0xE0002001	<b>E4RU_ERROR</b>	Ошибка выполнения функции
0xE0002002	<b>E4RU_INVALID_PARAMETER</b>	Неверный параметр
0xE0002003	<b>E4RU_INSUFFICIENT_BUFFER</b>	Нехватка буфера
0xE0002004	<b>E4RU_DEVICE_NOT_FOUND</b>	Ключ не найден
0xE0002005	<b>E4RU_FILE_TYPE_MISMATCH</b>	Неправильный тип файла
0xE0002006	<b>E4RU_MODULE_NO_CONTENT</b>	Модуль не содержит данные
0xE0002007	<b>E4RU_PACKAGE_NO_CONTENT</b>	Пакет обновления не содержит данные
0xE0002008	<b>E4RU_FILE_CREATE_SIZE_ERROR</b>	Размер создаваемого файла неверный
0xE0002009	<b>E4RU_FILE_NAME_RESERVED</b>	Имя файла зарезервировано
0xE0002301	<b>E4RU_USERPIN_ERROR</b>	Неправильный PIN код пользователя
0xE0002302	<b>E4RU_INVALID_EXECUTE_DIR</b>	Неправильный каталог
0xE0002303	<b>E4RU_INVALID_PACKAGE</b>	Ошибка пакета обновления
0xE0002304	<b>E4RU_DEVICE_ERROR_MAX</b>	Превышено максимальное количество ошибок
0xE0002305	<b>E4RU_DEVICE_NOT_ENOUGH_SPACE</b>	В ключе недостаточно свободного пространства
0xE0002306	<b>E4RU_FILE_OVER_THRESHOLD</b>	Превышено количество файлов
0xE0002307	<b>E4RU_MODULE_MAX</b>	Максимальное количество модулей превышено
0xE0002308	<b>E4RU_DEVICE_TYPE_ERROR</b>	Неправильный тип ключа или версия ниже требуемой
0xE0002309	<b>E4RU_PACKAGE_USED</b>	Пакет обновления уже использован
0xE0002310	<b>E4RU_NEW_MODULE_NOT_INSTALLED</b>	Новый модуль обновления не установлен
0xE0002311	<b>E4RU_INCORRECT_DEVICE_ID</b>	ID ключа не верное

## 4.7 Описание структуры SYSTEM\_INFORMATION

```
typedef struct _SYSTEM_INFORMATION {
    DWORD    GlobalSerial;
    DWORD    PrivateSerial;
    BYTE     MaxError;
    BYTE     ModuleCount;
    BYTE     ErrorNum;
    BYTE     RecordCount;
    WORD     MinorVersion;
    DWORD    FreeSpace;
    BYTE     DeviceId[8];
} SYSTEM_INFORMATION, *PSYSTEM_INFORMATION;
```

Таблица 2

Наименование поля	Тип	Описание
GlobalSerial	DWORD	Публичный серийный номер ключа.
PrivateSerial	DWORD	Приватный серийный номер ключа. Его необходимо проверять в случае создания и предоставления пакета обновления конкретному пользователю. Этот серийный номер нельзя подменить или обнулить.
MaxError	BYTE	01-FF   Счетчик возможных ошибок.
ModuleCount	BYTE	01-FE   Количество возможных модулей
ErrorNum	BYTE	00-FF   Количество произошедших ошибок при обновлении
RecordCount	BYTE	00-FE   Количество записей
MinorVersion	WORD	Версия установленного пакета обновления.
FreeSpace	DWORD	Количество свободного пространства в ключе
DeviceID[8]	BYTE	ID ключа. Это значение можно использовать при создании пакета обновления направленного только определённому устройству.

Информацию о номере/версии уже установленного обновления можно получить из ключа при помощи функции `e4ru_GetDeviceInfo()`, оно будет содержаться в структуре `SYSTEM_INFORMATION`, в параметре `GlobalSerial` (если установлен флаг `E4RU_FLAG_ALL_USER`) или `PrivateSerial` (если установлен флаг `E4RU_FLAG_INDIVIDUAL_USER`).

## 4.8 Описание структуры DEVICE\_MODULE\_INFO

```
typedef struct _DEVICE_MODULE_INFO {
    BYTE        Mid;
    DWORD       Version;
    BYTE        Valid;
} DEVICE_MODULE_INFO, *PDEVICE_MODULE_INFO;
```

Таблица 3

Наименование поля	Тип	Возможное значение	Описание
Mid	byte	01-FE	ID Модуля обновления
Version	dword	00-FFFFFFFF	Версия модуля обновления
Valid	byte	0 или 1	0: модуль обновления не установлен 1: модуль обновления установлен

## 4.9 Описание структуры PACKAGE\_INFORMATION

```
typedef struct _PACKAGE_INFORMATION {
    BYTE      Scope;
    BYTE      ValidType;
    BYTE      UpdateType;
    BYTE      DeviceID[8];
    DWORD     MajorVersion;
    WORD      MinorVersion;
} PACKAGE_INFORMATION, *PPACKAGE_INFORMATION;
```

Таблица 4

Наименование поля	Тип	Значения	
Scope	byte	<b>E4RU_FLAG_ALL_USER</b>	Пакет обновления может быть использован любым пользователем
		<b>E4RU_FLAG_INDIVIDUAL_USER</b>	Пакет обновления предназначен индивидуальному пользователю
ValidType	byte	<b>E4RU_FLAG_UPDATE_ONCE</b>	Разрешено использовать пакет обновления только один раз
		<b>E4RU_FLAG_UPDATE_PERMANENT</b>	Разрешено использовать пакет обновления многократно
UpdateType	byte	<b>E4RU_FLAG_CREATE_NEW_MODULE</b>	Разрешено создавать новые модули
		<b>E4RU_FLAG_UPDATE_EXISTING_ONLY</b>	Разрешено только обновлять существующие модули
DeviceID[8]	byte	ID ключа. Это значение можно использовать при создании пакета обновления направленного только определённому устройству.	
MajorVersion	dword	Старшее значение версии пакета обновления	
MinorVersion	dword	Младшее значение версии пакета обновления	

#### 4.10 Описание структуры PACKAGE\_MODULE\_INFO

```
typedef struct _PACKAGE_MODULE_INFO {
    BYTE        Mid;
    BYTE        Name[16];
    DWORD       Version;
    BYTE        EFNum;
} PACKAGE_MODULE_INFO, *PPACKAGE_MODULE_INFO;
```

Таблица 5

Наименование поля	Тип	Возможное значение	Описание
Mid	byte	01-FE	ID Модуля обновления
Name[16]	byte		Название модуля обновления
Version	dword	00-FFFFFFFF	Версия модуля обновления
EFNum	byte	01-FF	Количество файлов в модуле обновления.

## Где купить наши продукты.

### Российская Федерация

ЗАО «Секьюлэб»  
196240, Россия, Санкт-Петербург,  
ул. Кубинская, д.75/1, офис 301

тел/факс: +7-(812)-635-5080

E-MAIL: [lock@senselock.ru](mailto:lock@senselock.ru)

WEB: <http://senselock.ru>

ICQ: 408265904

Google-Talk: senselock.ru

## **История документа.**

12.10.07 - версия 1.0.4 .