Работа на iOS

Перечисление списка подключенных доступных КриптоПро CSP считывателей:

```
@interface CProReader : NSObject
@property (assign, readwrite) NSString* nickname;
@property (assign, readwrite) NSString* name;
@property (assign, readwrite) NSString* media;
@property (assign, readwrite) uint8_t flags;
-(void)dealloc;
-(CProReader*) initWithData:(uint8_t*)dataPtr;
@end

@implementation CProReader

@synthesize name;
@synthesize nickname;
@synthesize media;
@synthesize flags;

-(CProReader*) init {
    [super init];
    self.name = nil;
    self.nickname = nil;
    self.media = nil;
    return self;
}

-(CProReader*) initWithData: (uint8_t*)dataPtr {
    [super init];
    self.nickname = [[[NSString alloc] initWithBytes:dataPtr length:strlen((char*)dataPtr) encoding:
NSUTF8StringEncoding] autorelease];
    dataPtr+=1+[self.nickname length];
    self.name = [[[NSString alloc] initWithBytes:dataPtr length:strlen((char*)dataPtr) encoding:
NSUTF8StringEncoding] autorelease];
    dataPtr+=1+[self.name length];
    self.media = [[[NSString alloc] initWithBytes:dataPtr length:strlen((char*)dataPtr) encoding:
NSUTF8StringEncoding] autorelease];
    dataPtr+=1+[self.name length];
    self.flags = *dataPtr;
    return self;
}

-(void)dealloc {
    [super dealloc];
}
@end

static const int kGostProvType = 75;

NSArray* getReaderList()
{
    NSMutableArray* readerList = nil;

    DWORD error = ERROR_SUCCESS;
    HCRYPTPROV  hCryptProv = 0;
    CSP_BOOL    bResult = 0;
    DWORD       dwLen = 0;

    bResult = CryptAcquireContext(&hCryptProv, NULL, NULL, kGostProvType, CRYPT_VERIFYCONTEXT);
    if (!bResult) {
        error = CSP_GetLastError();
        NSLog(@"CryptAcquireContext(CRYPT_VERIFYCONTEXT): %x\n", error);
    }

    if(0 == hCryptProv) {
        NSLog(@"Invalid HCRYPTPROV");
        return nil;
    }
```

```
    BYTE cryptFirst = CRYPT_FIRST;

    for (;1;) {

        CSP_SetLastError(ERROR_SUCCESS);
        bResult = CryptGetProvParam(hCryptProv, PP_ENUMREADERS, NULL, &dwLen, CRYPT_MEDIA | cryptFirst);
        error = CSP_GetLastError();
        if (error == ERROR_NO_MORE_ITEMS)
            break;
        if (!bResult)
        {
            printf("CryptGetProvParam(PP_ENUMREADERS, LEN): %x\n", error);
            break;
        }



        NSMutableData* data = [[[NSMutableData alloc] initWithCapacity:dwLen] autorelease];

        CSP_SetLastError(ERROR_SUCCESS);
        bResult = CryptGetProvParam(hCryptProv, PP_ENUMREADERS, (BYTE*)[data bytes], &dwLen, CRYPT_MEDIA |
cryptFirst);
        cryptFirst = 0;
        error = CSP_GetLastError();
        if (error == ERROR_NO_MORE_ITEMS)
            break;
        if (!bResult)
        {
            printf("CryptGetProvParam(PP_ENUMREADERS, NAME): %x\n", error);
            break;
        }

        BYTE* dataPtr = (BYTE*)[data bytes];
        CProReader* reader = [[[CProReader alloc] initWithData:dataPtr] autorelease];

        if (nil == readerList) {
            readerList =[[NSMutableArray new] autorelease];
        }

        [readerList addObject: reader];
    }
    return readerList;
}
```

В случае если в возвращенном функцией getReaderList() массиве считывателей у объекта CProReader и убедиться, с именем name=@"Aktiv Rutoken ECP BT XXXXXXXX" поле media имеет значение @"rutoken_ecp_YYYYYYY" .

Требования к приложению

На приложение, обращающееся к Токену накладываются следующие требования:

- В Info.plist приложения должен быть указан протокол поддерживаемых внешних устройств (Supported external accessory protocols) со значением "com.aktivco.rutokenecp"

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    ...
    <key>UISupportedExternalAccessoryProtocols</key>
    <array>
    ...
        <string>com.aktivco.rutokenecp</string>
    ...
    </array>
...
</dict>
</plist>
```

- Приложение должно быть слинковано со следующими фреймворками:
    - RDRRtSupCp.framework – модуль поддержки считывателя Rutoken ECP BT для КриптоПро;
    - RtPKCS11ECP.framework – модуль, реализующий стандарт PKCS#11;
    - RtPcsc.framework – модуль поддержки PCSC-уровня;
    - стандартные фреймворки Security.framework, ExternalAccessory.Framework, Foundation.framework.

Примерный порядок действий пользователя при взаимодействии с приложением

Примерный порядок действий пользователя при первом запуске приложения поддерживающего установку защищённого канала с Токеном

1) пользователь запускает приложение

1.1.1) приложение сообщает пользователю что необходимо включить Bluetooth на устройстве (в случае, если радиоканал Bluetooth на устройстве был включён заранее, пункты 1.1.1-1.1.2 будут пропущены)

1.1.2) приложение показывает кнопку включения Bluetooth или перебрасывает на экран настроек Bluetooth

1.2.1) приложение сообщает пользователю что ему необходимо подключить Токен к устройству (показывает как надо нажать и подержать кнопку на Токене, пока он не заморгает)

1.2.2) Пользователь нажимает кнопку "подключить" и приложение его перебрасывает в настройки Bluetooth где он видит Токен в списке Bluetooth устройств в состоянии "без пары".

1.2.3) Пользователь нажимает на название устройства и ждёт пока произойдёт спаривание

1.2.4) Пользователь возвращается в приложение

2)Приложение находит токен и при перечислении считывателей посредством вызова CryptGetProvParam(PP_ENUMREADERS) по параметру szMedia="rutoken_ecp_XXXXXXXX" определяет, что все корректно.

3) Приложение сообщает пользователю что его Токен теперь может работать с этим приложением

4) Приложение работает с контейнером на Токене

Примерный порядок действий при повторном запуске приложения

1) пользователь запускает приложение

1.1.1) приложение сообщает пользователю что необходимо включить Bluetooth на устройстве (в случае, если радиоканал Bluetooth на устройстве был включён заранее, пункты 1.1.1-1.1.2 будут пропущены)

1.1.2) приложение показывает кнопку включения Bluetooth (iOS7) или перебрасывает на экран настроек Bluetooth

1.2.1) приложение сообщает пользователю что ему необходимо подключить Токен к устройству (показывает как надо нажать и подержать кнопку на Токене, пока он не заморгает синим светодиодом)

1.2.2) токен автоматически подключается к устройству

2) Приложение находит токен и при перечислении считывателей посредством вызова CryptGetProvParam(PP_ENUMREADERS) по параметру szMedia="rutoken_ecp_XXXXXXXX" корректность.

3) Приложение работает с контейнером на Токене