Especificação da API de comunicação do Módulo Criptográfico

Versão 1.7.1

01 de Julho de 2015

www.acura.com.br

Sumário

Tabela de Revisões

Versão da API	Data	Descrição
1.0	17/06/2014	Versão inicial
1.1	26/06/2014	Adicionado mais códigos de erro que podem ocorrer no firmware.
1.2	27/10/2014	Adicionado contexto na biblioteca. A API agora suporta múltiplos módulos ao mesmo tempo. Adicionado novos códigos de erro. Adicionado a checagem de assinatura ao enviar a chave de sessão ao MCR. Removido o método setTime. Alterado o nome do método getMCRPubKey para getPubKey.
1.4	01/04/2015	Corrigido uma falha na biblioteca que poderia causar “falha de segmentação” na execução do método getPubKey.
1.5	09/04/2015	Corrigido uma falha na função que envia a lista de chaves de grupo para o MCR. Adicionado um método, setLog, para ativar ou desativar a gravação de mensagens de depuração em arquivo. Removido a dependência de versionamento dos símbolos referentes às funções da biblioteca OpenSSL.
1.7.1	01/07/2015	A biblioteca agora é *thread-safe*, ou seja, os métodos da API podem ser chamados de *threads* diferentes. O método sendGidList agora importa apenas as

		chaves que o módulo criptográfico ainda não possui.

Lista de Tabelas

Abreviações

API	*Application Programming Interface.*
CRC-16	Código do tipo CRC de 16 bits de tamanho, com polinômio conforme definido na norma ISO 13239, também conhecido como CRC-16-CCITT.
EGC	Entidade Gestora de Chaves.
GID	*Group Identifier* – Identificador de Grupo.
MCR	Módulo criptográfico.
RSA	Algoritmo de criptografia de chave pública.
TAG	Etiqueta eletrônica de RFID.
USB	*Universal Serial Bus.*

1. Introdução

Neste documento são descritas as funções disponibilizadas pela API de comunicação do módulo criptográfico (MCR) de forma a facilitar a integração desta API às aplicações interessadas em utilizar o módulo. Esta API foi desenvolvida em linguagem C e é fornecida como uma biblioteca dinâmica, em .dll para Windows e .so para Linux. A biblioteca está disponível para as arquiteturas x86, x64, ARMv6 e ARMv7.

2. Requisitos mínimos

Computador com Sistema Operacional Windows XP ou superior, Linux 2.6.x ou superior. Processador de 700MHz ou superior de 32 ou 64 bits, 512 MB de RAM, porta USB 1.1. Como a API depende da biblioteca *libusb* para se comunicar com o MCR, é necessário ter instalado o pacote *libusb-win32* versão 1.2.6.0 ou superior se for utilizado Windows ou o pacote *libusb* versão 1.0.18 ou superior se for utilizado Linux. É necessário ter também instalado o pacote *OpenSSL* versão 1.0.1h ou superior.

3. Definições

3.1. Tipos de variável

TIPO	DESCRIÇÃO
uint8_t	Variável de 8 bits sem sinal
uint16_t	Variável de 16 bits sem sinal
uint32_t	Variável de 32 bits sem sinal

3.2. Códigos de retorno

Tabela 1: Erros ocorridos na API.

NOME	VALOR	DESCRIÇÃO
LIBMCR_OK	0x0000	Sucesso.
LIBMCR_ERR_LIBRARY_NOT_INIT	0xE001	Biblioteca não inicializada.
LIBMCR_ERR_NULL_PARAMS	0xE002	Parâmetro inválido.
LIBMCR_ERR_INV_KEYLEN	0xE004	Tamanho inválido da chave de sessão.
LIBMCR_ERR_KEY_FOPEN	0xE005	Erro ao abrir o arquivo da chave pública.
LIBMCR_ERR_MALLOC	0xE006	Erro ao tentar alocar memória.
LIBMCR_ERR_GID_NOT_FOUND	0xE007	GID não encontrado.
LIBMCR_ERR_INV_LISTSIZE	0xE008	Tamanho inválido da lista de chaves.

LIBMCR_ERR_DEVICE_NOT_FOUND	0xE009	MCR não encontrado.
LIBMCR_ERR_BUF_TOO_SMALL	0xE00A	Buffer muito pequeno.
LIBMCR_ERR_PTHREAD	0xE00B	Erro ao criar thread.
LIBMCR_ERR_NO_SPACE	0xE00C	Não há espaço suficiente no MCR para armazenar a lista de chaves.
LIBMCR_ERR_RSA_PUB_READ	0xE00D	Erro ao ler a chave pública do MCR.
LIBMCR_ERR_INV_CTX	0xE00F	Contexto inválido.
LIBMCR_ERR_MAX_CTX	0xE010	API não pode alocar contexto, pois o número máximo de contextos foi atingido.
LIBMCR_ERR_CRYPT_SIZE_TOO_BIG	0xE011	Bloco criptográfico muito grande (>256bytes).
LIBMCR_ERR_NO_NEW_KEYS	0xE012	Não há novas chaves a serem importadas.

Tabela 2: Erros ocorridos no firmware.

NOME	VALOR	DESCRIÇÃO
FWMCR_ERR_GENERIC	0xA000	Erro genérico.
FWMCR_ERR_WRAPPING_KEY	0xA001	Erro ao descriptografar a chave de sessão ou a lista de chaves.
FWMCR_ERR_KEY_NOT_FOUND	0xA002	Chave não encontrada.
FWMCR_ERR_USER_NOT_FOUND	0xA003	Usuário não encontrado.
FWMCR_ERR_CMD_NOT_SUPPORTED	0xA004	Comando não suportado.
FWMCR_ERR_SIG_INVALID	0xA005	Assinatura inválida.

4. Funções da API

Esta seção descreve e detalha todas as funções que são exportadas pela biblioteca de comunicação do MCR. Todas as funções descritas nesta seção possuem o seguinte formato: *uint32_t função(param_1, param_2, param_3, ...)* onde *param_1* é o primeiro parâmetro descrito na lista de parâmetros, *param_2* é o segundo parâmetro descrito na lista de parâmetros e assim por diante.

4.1. initLibrary

- Descrição: Função para inicializar a biblioteca e se conectar ao MCR. Deve ser a primeira função a ser chamada antes de qualquer outra. Esta função aloca um contexto e retorna ao usuário o índice deste contexto.
- Requisitos:
  - O MCR deve estar conectado na porta USB antes que esta função seja chamada.
  - A aplicação deve ter permissão de leitura/escrita ao dispositivo USB.
- Parâmetros:
  - uint8_t * ctxIndex: Ponteiro para a variável que irá armazenar o índice do contexto.
- Retorno: LIBMCR_OK, LIBMCR_ERR_DEVICE_NOT_FOUND, LIBMCR_ERR_PTHREAD, LIBMCR_ERR_MAX_CTX, LIBMCR_ERR_NULL_PARAMS, LIBMCR_ERR_MALLOC

4.2. closeLibrary

- Descrição: Função para desconectar o MCR do barramento USB.
- Requisito: A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
- Retorno: LIBMCR_OK, LIBMCR_ERR_LIBRARY_NOT_INIT, LIBMCR_ERR_INV_CTX

4.3. userLogin

- Descrição: Função para logar o usuário de operação no MCR.
- Requisito: A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
- Retorno: LIBMCR_OK, LIBMCR_ERR_LIBRARY_NOT_INIT, FWMCR_ERR_GENERIC, LIBMCR_ERR_INV_CTX

Observação: É necessário chamar esta função antes de utilizar pela primeira vez a função de descriptografia *(decrypt)*.

4.4. userLogout

- Descrição: Função para deslogar o usuário de operação do MCR.
- Requisito: A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.

- Retorno: LIBMCR_OK, LIBMCR_ERR_LIBRARY_NOT_INIT, FWMCR_ERR_GENERIC, LIBMCR_ERR_INV_CTX

4.5. sendSessionKey

- Descrição: Função para enviar a chave de sessão recebida da EGC ao MCR.
- Requisito: A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
  - uint8_t *key: Ponteiro para a chave de sessão recebida da EGC.
  - uint32_t size: Tamanho da chave, em bytes.
  - uint8_t *signature: Ponteiro para a assinatura recebida da EGC.
  - uint16_t sigSize: Tamanho da assinatura, em bytes.
  - uint16_t crc16: CRC-16 da chave, valor também recebido da EGC.
- Retorno: LIBMCR_OK, LIBMCR_ERR_NULL_PARAMS, LIBMCR_ERR_INV_KEYLEN, LIBMCR_ERR_LIBRARY_NOT_INIT, FWMCR_ERR_GENERIC, FWMCR_ERR_WRAPPING_KEY, LIBMCR_ERR_INV_CTX, FWMCR_ERR_SIG_INVALID

Observação: A chave de sessão recebida da EGC está em formato base64. Antes de chamar esta função deve-se decodificar a chave para o formato binário. O mesmo é válido para o campo da assinatura.

4.6. getPubKey

- Descrição: Função que retorna a chave pública RSA-2048 do MCR.
- Requisito: A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
  - const uint8_t *filename: Nome do arquivo onde a chave pública será armazenada.
- Retorno: LIBMCR_OK, LIBMCR_ERR_NULL_PARAMS, LIBMCR_ERR_KEY_FOPEN, LIBMCR_ERR_LIBRARY_NOT_INIT, FWMCR_ERR_GENERIC, LIBMCR_ERR_INV_CTX

Observações: Se o arquivo já existir ele será sobrescrito. A aplicação deve ter permissão de leitura/escrita no diretório onde o arquivo será salvo.

4.7. sendGidList

- Descrição: Função para enviar a lista de chaves recebida da EGC para o MCR. Chamadas subsequentes a esta função irá importar apenas as chaves novas e que ainda não foram enviadas ao MCR.

- Requisitos:
  - A biblioteca deve estar inicializada.
  - O MCR deve conter uma chave de sessão.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
  - uint32_t gids[]: Lista de GIDs recebida da EGC.
  - uint8_t **keys: Lista de chaves recebida da EGC.
  - uint16_t crc16[]: Lista de CRC-16 de cada chave, recebida da EGC.
  - uint32_t listsize: Tamanho das listas. As listas têm tamanhos iguais.
- Retorno: LIBMCR_OK, LIBMCR_ERR_NULL_PARAMS, LIBMCR_ERR_KEY_FOPEN, LIBMCR_ERR_LIBRARY_NOT_INIT, LIBMCR_ERR_NO_SPACE, FWMCR_ERR_GENERIC, FWMCR_ERR_WRAPPING_KEY, LIBMCR_ERR_INV_CTX, LIBMCR_ERR_NO_NEW_KEYS

Observações: A lista de chaves recebida da EGC está em formato base64. Antes de chamar esta função deve-se decodificar a lista de chaves para o formato binário. A posição 0 da lista de chaves corresponde ao GID na posição 0 da lista de GIDs e ao CRC-16 na posição 0 da lista de CRC-16, e assim por diante.

4.8. decrypt

- Descrição: Função para descriptografar o bloco cifrado da TAG.
- Requisitos:
  - A biblioteca deve estar inicializada.
  - O MCR deve conter a lista de chaves.
  - O usuário de operação deve estar logado no MCR.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
  - uint32_t gid: GID da TAG.
  - uint8_t *cryptedData: Ponteiro para o bloco cifrado da TAG.
  - uint32_t cryptedDataSize: Tamanho do bloco cifrado.
  - uint8_t *decryptedData: Ponteiro para um buffer que irá conter o bloco decifrado.
  - uint32_t *decryptedDataSize: Tamanho do buffer que irá conter o bloco decifrado.
- Retorno: LIBMCR_OK, LIBMCR_ERR_NULL_PARAMS, LIBMCR_ERR_GID_NOT_FOUND, LIBMCR_ERR_LIBRARY_NOT_INIT, LIBMCR_ERR_MALLOC, LIBMCR_ERR_BUF_TOO_SMALL, FWMCR_ERR_GENERIC, LIBMCR_ERR_INV_CTX, LIBMCR_ERR_CRYPT_SIZE_TOO_BIG

Observações: O buffer fornecido para esta função precisa ter tamanho suficiente para armazenar o bloco decifrado. Após chamar esta função, e caso o buffer tenha tamanho suficiente, a variável *decryptedDataSize* irá conter o tamanho exato do bloco decifrado.

4.9. setLog

- Descrição: Função para ativar ou desativar a gravação de mensagens de depuração no arquivo local de log.
- Requisitos:
  - A biblioteca deve estar inicializada.
- Parâmetros:
  - uint8_t ctxIndex: Índice do contexto.
  - uint8_t enable: Ativa a gravação de mensagens de depuração quando enable = 1, desativa quando enable = 0
- Retorno: LIBMCR_OK, LIBMCR_ERR_LIBRARY_NOT_INIT, LIBMCR_ERR_MAX_CTX, LIBMCR_ERR_INV_CTX

Observações: O arquivo de log é gravado na mesma pasta da aplicação, com o seguinte nome: libmcr.log.%Número do contexto%. Por exemplo, se o contexto atribuído for igual a 0, o arquivo de log terá o seguinte nome: libmcr.log.0.