PMA16329-AA

DRAFT - RESTRICTED

SpringCard Smart Readers & RFID Scanners

Specification of Master Cards

www.springcard.com

Document Identification

Category	Admin/Config Manual		
Family/Customer	Smart Readers & RFID Scanners		
Reference	PMA16329	Version	AA
Status	Draft	Classification	Restricted
Keywords	RDR, Prox'N'Roll RFID Scanner, FunkyGate, Prox'N'Drive RFID Scanner, Configuration, Desfire, Master Card		
Abstract			

File name	V:\Dossiers\SpringCard\A-Notices\RFID scanners et lecteurs\IWM2-Commun\[PMA16329-AA] RDR and RFID Scanners - Specification of Master Cards.odt		
Date saved	19/09/16	Date printed	03/06/14

Revision History

Ver.	Date	Author	Valid. by		Approv. by	Details
			Tech.	Qual.		
AA	19/09/16	JDA				Created from internal documentations

Contents

1. Introduction

1.1. Abstract

SpringCard offers Access Control Readers, OEM Readers and PC-connected Readers dedicated to the automated processing of contactless smart cards and RFID labels or tags:

- The **FunkyGate** family: wall-mounted access control readers, available with either Data+Clock, Wiegand, serial RS-232, serial RS-485, serial emulation on top of USB, and TCP over Ethernet communication options,
- The **RFID Scanner** family: USB products for the desktop operating in keyboard emulation mode ("wedge")
- The **RDR** family: a wide range of OEM Readers featuring various communication options (RS-232, RS-TTL, RS-485, serial emulation on top of USB).

All these families share a large part of their feature, one of them being the ability to get their configuration settings on-the-field, thanks to a **secured, contactless Master Card**. Master Cards are created using **MultiConf**, an application running on Microsoft Windows and provided free-of-charge by SpringCard. MultiConf uses any SpringCard PC/SC Coupler to encode the Master Cards.

This document is the public specification of the Master Card feature, as it is implemented in all SpringCard Readers since version 1.40. It is provided as a mean for security experts to evaluate the system, or for third-party application developers to create an alternative to MultiConf.

> *The value of the two factory keys are missing from this document. They could be obtained from SpringCard only after signing a Non-Disclosure Agreement.*

1.2. Audience

This manual is designed for use by application developers and system integrators. It assumes that the reader has a good knowledge of computer development and security schemes.

1.3. Support and updates

Useful related materials (product datasheets, application notes, sample software, HOWTOs and FAQs…) are available at SpringCard's web site:

www.springcard.com

Updated versions of this document and others are posted on this web site as soon as they are available.

For technical support enquiries, please refer to SpringCard support page, on the web at

www.springcard.com/support

1.4. Related documents

This document uses concepts and vocabulary defined in the documentation of the "Templates", the system that allows a SpringCard Reader to process a large family of cards.

Document ref.	Content
PMA13205	Smart Readers and RFID Scanners Template System

2. Definition of Master Cards

2.1. Supported cards

First generation of Master Cards were NXP Desfire EV0 4K.

NXP Desfire EV1 2K, NXP Desfire EV1 4K and NXP Desfire EV1 8K are acceptable replacements.

> *The security algorithm rely on the card's UID. Random-ID configuration of the Desfire cards is therefore not supported.*

2.2. Application and files

The Master Card application uses Desfire AID $_{h}$504143.

The application has two files:

- File $_{h}$01 stores the configuration to be written into the Reader. The size of this file must be 512 bytes, exactly.
- File $_{h}$02 stores the digital signature that proves to the Reader that the data come from a valid source. The size of this file must be 16 bytes, exactly.

2.3. Authentication and access rules

The Reader gets authenticated onto the application using key #0. The authentication method is Desfire Legacy 3DES (2K).

The Reader uses a key diversification algorithm to compute key #0 based on a Master Key and the card's UID:

- Let *MasterAuthKey* be a 16-B-long Master Key for Authentication,
- Let *CardUID* be the 7-B-long card's UID,
- Compute *CardAuthKey* = HMAC-MD5 (*MasterAuthKey*, *CardUID*)

The card shall be formatted with key #0 = *CardAuthKey*.

Both file $_{h}$01 and file $_{h}$02 shall be readable in ciphered mode (Desfire Comm. Mode = 3) after authentication with key #0.

2.4. Content of file ${}_{H}$01

File ${}_{h}$01 stores the configuration to be written into the Reader. The data bytes in file ${}_{h}$01 uses the T,L,V (Tag, Length, Value) representation:

- Tag is the address of the register to be written (for instance, ${}_{h}$60 for the OPT register, ${}_{h}$20 for the LKL register of Template #2, etc),
- Length is the length of the Value field, expressed in bytes. Max length is 32 (${}_{h}$20). A Master Card holding a T,L,V with L>32 will be rejected. Setting length to ${}_{h}$00 deletes the current value of a register (the register retrieves its default, out-of-factory value),
- Value is the content of the register.

There are a two special values:

1. Tag = ${}_{h}$FF, Length = 0 means “erase” (all registers retrieves their default, out-of-factory value). This special value should be the first entry in the file,
2. Tag = ${}_{h}$FF, Length = 7 writes a Mifare Classic key into the Reader’s Micore chipset. The 1st byte of the Value field is the address of the key, and the next 6 bytes its actual value,

Following the set of configuration entries, file ${}_{h}$01 must be filled-up by ${}_{h}$00.

2.5. Content of file ${}_{H}$02

File ${}_{h}$02 stores the digital signature that proves to the Reader that the data come from a valid source.

The signature algorithm is a HMAC, implemented as follow:

- Let *Content* be the content of file ${}_{h}$01 (including the h00 bytes following the actual data; *Content* is exactly 512-B long),
- Let *MasterSignKey* be the 16-B-long Master Key for Signature,
- Let *CardUID* be the 7-B-long card’s UID,
- Compute *CardSignKey* = HMAC-MD5 (*MasterSignKey*, *CardUID*)
- Compute *Sign* = HMAC-MD5 (*CardSignKey*, *Content*)

The card shall be encoded with content of file ${}_{h}$02 = *Sign*.

3. Keys used by the Reader

3.1. User-defined Keys

As all the other Reader’s settings, the Keys to secure the Master Cards are defined by some configuration registers.

3.1.1. Authentication Key

Configuration register ${}_{h}$55 stores *MasterAuthKey* together with the key number and a few options. The format of the register is the same as register **AUT for Desfire, authentication EV0**, as define in **PMA13205**.

Specification of register ${}_{h}$55 (size 17B)

Bits	Value	Meaning
Byte 0		
		Communication mode[1]
7-6	${}_{b}$00	Plain
	${}_{b}$01	MACed with using the session key
	${}_{b}$10	*RFU*
	${}_{b}$11	Encrypted using the session key
		Key diversification algorithm
5-4	${}_{b}$00	No diversification
	${}_{b}$01	Diversification using NXP RC171 algorithm
	${}_{b}$10	Diversification using HMAC-MD5
	${}_{b}$11	*RFU*
		Key number within the Desfire Master Card application
3-0	${}_{b}$0000 to ${}_{b}$1110	Must be ${}_{b}$0000 (MasterCard uses key # 0)
Bytes 1 to 16		
Value of the DES or 3-DES *MasterAuthKey* (16 bytes) For a DES key, both halves of the key are equal.		

(Mandatory values are shown in red. Choosing another value leads to unexpected behaviour and is not supported).

[1] The same communication mode applies to both file ${}_{h}$01 and file ${}_{h}$02.

3.1.2. Signature Key

Configuration register ${}_{h}$56 stores *MasterSignKey* together with a few options.

Specification of register ${}_{h}$56 (size 17B)

Bits	Value	Meaning
Byte 0		
Communication mode		
7-6	${}_{b}$00	*RFU, must be ${}_{b}$00*
Key diversification algorithm		
5-4	${}_{b}$00	No diversification
	${}_{b}$01	Diversification using NXP RC171 algorithm
	${}_{b}$10	Diversification using HMAC-MD5
	${}_{b}$11	*RFU*
Key # within the Desfire Master Card application		
3-0	${}_{b}$0000	*RFU, must be ${}_{b}$0000*
Bytes 1 to 16		
Value of *MasterSignKey* (16 bytes)		

(Mandatory values are shown in red. Choosing another value leads to unexpected behaviour and is not supported).

3.2. Factory Keys

When a register is not explicitly configured, the Reader uses its default, factory-defined value.

For register ${}_{h}$55, factory value is ${}_{h}$E0 *<Factory MasterAuthKey>.*

For register ${}_{h}$56, factory value is ${}_{h}$20 *<Factory MasterSignKey>.*

Both values could be disclosed under NDA.

Disclaimer

This document is provided for informational purposes only and shall not be construed as a commercial offer, a license, an advisory, fiduciary or professional relationship between PRO ACTIVE and you. No information provided in this document shall be considered a substitute for your independent investigation.

The information provided in document may be related to products or services that are not available in your country.

This document is provided "as is" and without warranty of any kind to the extent allowed by the applicable law. While PRO ACTIVE will use reasonable efforts to provide reliable information, we don't warrant that this document is free of inaccuracies, errors and/or omissions, or that its content is appropriate for your particular use or up to date. PRO ACTIVE reserves the right to change the information at any time without notice.

PRO ACTIVE doesn't warrant any results derived from the use of the products described in this document. PRO ACTIVE will not be liable for any indirect, consequential or incidental damages, including but not limited to lost profits or revenues, business interruption, loss of data arising out of or in connection with the use, inability to use or reliance on any product (either hardware or software) described in this document.

These products are not designed for use in life support appliances, devices, or systems where malfunction of these product may result in personal injury. PRO ACTIVE customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify PRO ACTIVE for any damages resulting from such improper use or sale.

Copyright notice

Editor's information

PRO ACTIVE SAS company with a capital of 227 000 €

RCS EVRY B 429 665 482

Parc Gutenberg, 2 voie La Cardon

91120 Palaiseau – FRANCE

Contact information

For more information and to locate our sales office or distributor in your country or area, please visit

www.springcard.com