**PMD17128-AB**
- PUBLIC

# SpringBlue ID - BLE/NFC Identification Solution

## Developer's Implementation Manual

## Document Identification

| | | | |
|---|---|---|---|
| Category | Developer's manual | | |
| Family/Customer | SpringBlue | | |
| Reference | PMD17128 | Version | AB |
| Status | | Classification | Public |
| Keywords | SpringBlue, BLE, NFC | | |
| Abstract | | | |

| | | | |
|---|---|---|---|
| File name | V:\Dossiers\SpringCard\Notices\RFID scanners et lecteurs\SpringBlue\SpringBlue-Developer's Implementation Manual.odt | | |
| Date saved | 13/01/22 | Date printed | 05/12/12 |

## Revision History

| Ver. | Date | Author | Valid. by | | Approv. by | Details |
|---|---|---|---|---|---|---|
| | | | Tech. | Qual. | | |
| AA | 25/04/17 | JDA | | | | Creation |
| AB | 13/01/2022 | MBA | | | | Chapter 6.3.1 and 6.3.2 updated. |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

## Contents

# 1. Introduction

## 1.1. Abstract

SpringBlue ID is an innovative identification scheme, targeting smartphone applications, where a user's identification number is pushed securely to a SpringBlue ID-enabled Reader, either through NFC Host Card Emulation (HCE) or through Bluetooth Smart, *aka* BLE (Bluetooth Low Energy).

SpringBlue ID has been designed not only with security but also with privacy in mind: only the site owning a user ID is able to read it. SpringBlue ID opens numerous use cases related to user identification: physical access control, loyalty programs, car park, car sharing or bike sharing schemes, and more.

This document is the reference guide for developers who need to implement the SpringBlue ID scheme in their smartphone application. It presents the SpringBlue ID data model, the secure transaction, and details the exchanges between the smartphone and Reader through either BLE or HCE.

## 1.2. Audience

This manual is designed for use by application developers. It assumes that the reader has expert knowledge of computer development and a basic knowledge of the BLE and NFC communication standards, and of the ISO 7816-4 standard for smartcards.

## 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

# 2. Terms and definitions

## 2.1. Definitions

For the purposes of the document, the following terms and definitions apply:

**SpringBlue ID Reader or SBIR**

An electronic device that implements the Reader part of the SpringBlue ID transaction, over BLE or NFC, or both.

**SpringBlue ID Object or SBIO**

A smartphone or any other mobile electronic device that stores a SpringBlue ID and implements the User part of the transaction. The Object is identified by a 16-byte pseudo-unique ObjectID.

**Site** An instance of the SpringBlue ID scheme, identified by a 4-byte SiteID. A SpringBlue ID Reader is only able to read UserIDs belonging to the same Site. The SiteID is attributed by SpringCard to its customers (one SiteID per customer or per installation depending on the use case).

**User** A user belonging to a Site, identified by a 8-byte UserID. The UserID is attributed by the implementer. The implementer shall ensure that the attributed UserIDs are unique within a Site.

## 2.2. Abbreviations

**SOIK** Site’s ObjectID Key. The site-wide global AES key used to cipher the ObjectID between the SBIO and the SBIR.

**OSUK** Object + Site’s UserID Key. The AES key used to cipher the UserID cryptogram between the SBIO and the SBIR. This key is specific to the object (and to the site).

**MSUK** Master Key to protect Site’s UserIDs. The AES keys used to compute the OSUK for a given object (and site).

## 2.3. Glossary

**AES** Symbol for Advanced Encryption Standard (as defined in ISO/IEC 18033-3:2010),

| | |
|---|---|
| | symmetrical cryptographic algorithm using 128-bit data and key. |
| **AES-ECB** | AES Electronic Codebook mode – refers to a single-block AES operation |
| **ATR** | Answer To Reset (data returned by a smart card during startup) |
| **RFU** | Symbol for “Reserved for Future Use” |
| **SAM** | Secure Application Module |

# 3. The SpringBlue transaction

## 3.1. Basis

After the initial BLE/NFC discovery procedure (which will be detailed in the next chapters), The SpringBlue transaction relies on no more than 3 APDU (Illustration 1).

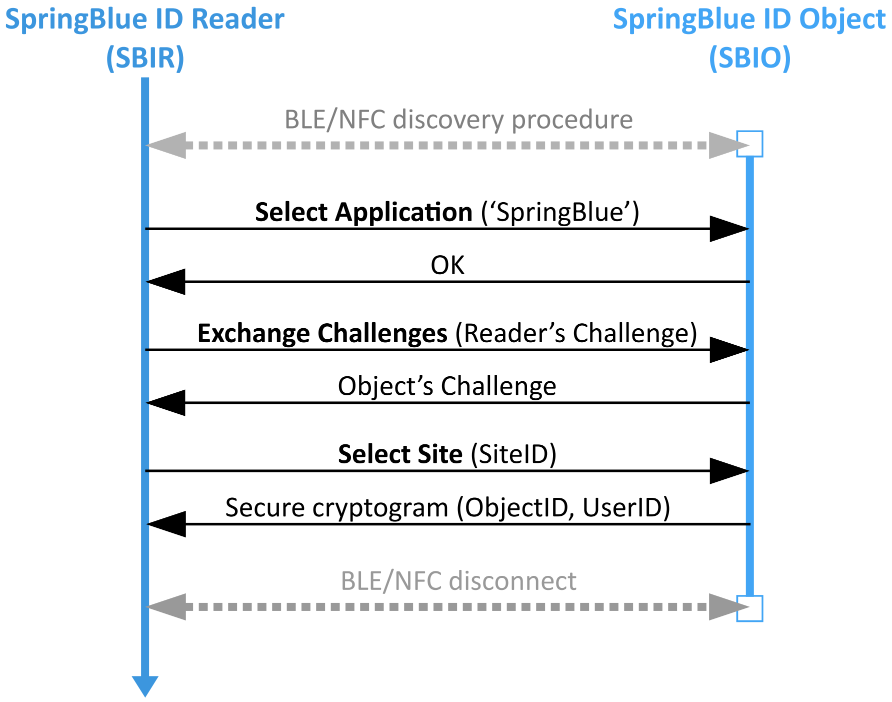
SpringBlue ID Reader
(SBIR)
SpringBlue ID Object
(SBIO)
BLE/NFC discovery procedure
**Select Application** ('SpringBlue')
OK
**Exchange Challenges** (Reader's Challenge)
Object's Challenge
**Select Site** (SiteID)
Secure cryptogram (ObjectID, UserID)
BLE/NFC disconnect

**Illustration 1: The SpringBlue transaction**

## 3.2. Select Application

On the SBIO's side, this 1st exchange allows the object's operating system to activate the application. The SpringBlue application running in the SBIO shall reset its state machine.

If no SBIO application is found in the object, the SBIR is notified of the error and closes the communication.

### 3.2.1. Select Application Command

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| CLA | $_h$00 | 1 | ISO/IEC 7816-4: SELECT Instruction, direct selection by DF Name |
| INS | $_h$A4 | 1 | |
| P1 | $_h$04 | 1 | |
| P2 | $_h$00 | 1 | |
| $L_C$ | $_h$10 | 1 | Length of the DF Name |
| DataIn | $_h$A0 $_h$00 $_h$00 $_h$06 $_h$14 $_h$53 $_h$70 $_h$72 $_h$69 $_h$6E $_h$67 $_h$42 $_h$6C $_h$75 $_h$65 $_h$30 | 16 | DF Name: SpringCard's registered application provider ID + ID of the SpringBlue application |
| $L_E$ | $_h$00 | 0 or 1 | (optional) |

### 3.2.2. Select Application Response

#### *a. Success*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| DataOut | | 0 - ? | Don't care |
| SW | $_h$90 $_h$00 | 2 | Status Word – success |

#### *b. Error*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| SW | $_h$6x $_h$xx | 2 | Status Word – error. See § 3.5. |

## 3.3. Exchange Challenges

This 2nd exchange has two roles:

1. Transmit to the SBIO the SBIR's Challenge (random number) that will be used to secure the UserID in the 3rd exchange,
2. Transmit to the SBIR the SBIO's Challenge (random number) that will be used to protect the ObjectID in the 3rd exchange.

### 3.3.1. Exchange Challenges Command

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| CLA | ${}_h$00 | 1 | ISO/IEC 7816-4 default CLA |
| INS | ${}_h$86 | 1 | Custom Exchange Challenges Instruction |
| P1 | ${}_h$00 | 1 | RFU – must be ${}_h$00 |
| P2 | ${}_h$00 | 1 | RFU – must be ${}_h$00 |
| $L_C$ | ${}_h$08 | 1 | Length of the data |
| DataIn | SBIR’s Challenge | 8 | SBIR’s Challenge on 8 bytes |
| $L_E$ | ${}_h$00 | 0 or 1 | (optional) |

### 3.3.2. Exchange Challenges Response

#### *a. Success*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| DataOut | SBIO’s Challenge | 8 | SBIO’s Challenge on 8 bytes |
| SW | ${}_h$90 ${}_h$00 | 2 | Status Word – success |

#### *b. Error*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| SW | ${}_h$6x ${}_h$xx | 2 | Status Word – error. See § 3.5. |

## 3.4. Select Site

This 3rd and last exchange has two roles:

1. Tell the SBIO which company or scheme the SBIR belongs to (SiteID parameter).
2. Return the SBIO's ObjectID and UserID to the SBIR in a secure cryptogram.

To address privacy concerns, the SBIO shall always return a "success" response. If the SBIO doesn't have a valid SiteID/UserID record for the SiteID selected by the SBIR, it shall generate a random response of the expected length (32 bytes).

### 3.4.1. Select Site Command

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| CLA | $_h$00 | 1 | ISO/IEC 7816-4: SELECT Instruction, select child DF |
| INS | $_h$A4 | 1 | |
| P1 | $_h$01 | 1 | |
| P2 | $_h$00 | 1 | |
| $L_C$ | $_h$04 | 1 | Length of the data |
| DataIn | SiteID | 4 | SiteID on 4 bytes |
| $L_E$ | $_h$00 | 0 or 1 | (optional) |

### 3.4.2. Select Site Response

#### a. *Success – SiteID/UserID record exists in the SBIO*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| DataOut | Secure cryptogram | 32 | See chapter 4. for details |
| SW | $_h$90 $_h$00 | 2 | Status Word – success |

#### b. *Success – No such SiteID/UserID record in the SBIO*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| DataIn | Random data | 32 | |
| SW | $_h$90 $_h$00 | 2 | Status Word – success |

### *c. Error*

| *Field* | *Value* | *Size* | *Description* |
|---|---|---|---|
| SW | $_h$6x $_h$xx | 2 | Status Word – error. See § 3.5. |

## 3.5. Listing of Status Words

The SpringBlue application running in the SBO may returns only Status Words taken from the list below. The operating system of the SBO is likely to return different Status Words if the application is not present or not reachable.

| *SW1* | *SW2* | *Meaning* |
|---|---|---|
| $_h$90 | $_h$00 | Success |
| $_h$67 | $_h$00 | Wrong length ($L_C$ is not coherent with DataIn) |
| $_h$69 | $_h$85 | Condition of use not satisfied<br>(example: no Exchange Challenge before Select Site) |
| $_h$6A | $_h$80 | Incorrect parameters in DataIn |
| $_h$6B | $_h$00 | Wrong parameter P1-P2 |
| $_h$6C | $_h$00 | Wrong length $L_E$ (present and not $_h$00) |
| $_h$6D | $_h$00 | INStruction not supported |
| $_h$6E | $_h$00 | CLAss not supported |

# 4. Secure transmission of the ObjectID and UserID

## 4.1. Basis

The SBIO's response to the SBIR's Select Site command is a secure cryptogram containing the ObjectID and the UserID. Only a genuine SBIR (i.e. a SBIR knowing the Site's MSUK and SOIK) is able to recover these data.

The construction of this response is shown in Illustration 2 below.

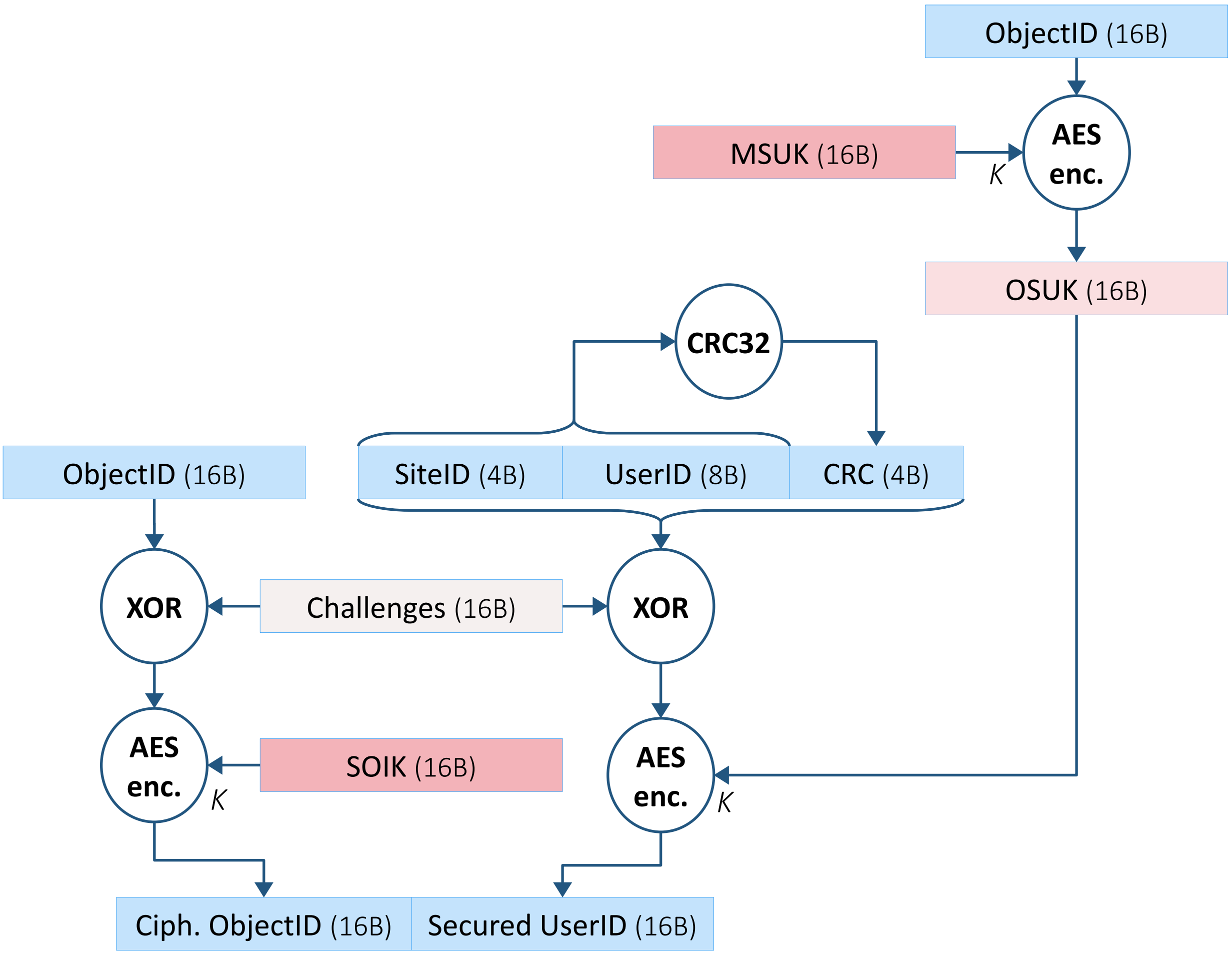

**Illustration 2: Construction of the Secure cryptogram**

*NB: The SBIO knows only its OSUK. The SBIR computes OSUK from MSUK and the ObjectID.*

## 4.2. Implementation in the SBIR

### 4.2.1. SBIR's constants

The SBIR has no constant value involved in the transaction.

### 4.2.2. SBIR's configuration data

The SBIR stores one configuration triplet consisting of

- SiteID,
- Site's ObjectID Key (SOIK),
- Master Key to protect Site's UserIDs (MSUK).

*The SOIK and MSUK are stored in the SBIR's SAM. All AES operations are performed within the SAM.*

### 4.2.3. Transaction handling

When the SBIR connects to a SBIO and starts a transaction, the SBIR runs the following algorithm:

1. Generate an 8-byte **SBIR's Challenge**, transmit it to the SBIO, and retrieve the 8-byte **SBIO's Challenge**,
2. Assemble the 16-byte value **Challenges = SBIR's Challenge .. SBIO's Challenge**,
3. Transmit its **SiteID** to the SBIO, and receive the 32-byte Secure cryptogram from the SBIO in response,
4. Decipher the first half of the Secure cryptogram using **SOIK** (AES ECB decrypt),
5. XOR this deciphered first half with **Challenges** to retrieve **ObjectID**,
6. Compute SBIO's **OSUK** by ciphering the **ObjectID** with **MSUK** (AES ECB encrypt),
7. Decipher the second half of the Secure cryptogram using this computed **OSUK** (AES ECB decrypt),
8. XOR this deciphered second half with **Challenges** to retrieve **SiteID .. UserID .. CRC**,
9. Verify that the **SiteID** received in the deciphered buffer is equal to the expected **SiteID**,
10. Verify that the **CRC** in the deciphered buffer is valid,
11. Extract the **UserID** from the deciphered buffer – and forward it to the downstream system.

*The pseudo-unique ObjectID is never exposed to the downstream system. Only the UserID is significant.*

## 4.3. Implementation in the SBIO

### 4.3.1. SBIO's constants

The SBIO is identified by one constant: the ObjectID, a 16-byte value.

The ObjectID must be physically associated to one particular device, tying every diversified key (OSUK) to a very object.

The implementer is responsible for providing a pseudo-unique ObjectID for the objects he releases. The ObjectID doesn't pretend to be unique, but collisions should be made as unlikely as possible.

A possible algorithms to do so could be:

**ObjectID = Hash ( Implementer-defined seed .. Mobile phone's IMEI )**

Any cryptographic hash function could be used: MD5, SHA-1, SHA-256… MD5 provides a 16-B output, for the other functions, the output has to be truncated.

### 4.3.2. SBIO's configuration data

The SBIO stores configuration quartets consisting of

- SiteID,
- Site's ObjectID Key (SOIK),
- Object + Site's UserID Key (OSUK).
- UserID.

The SBIO may store any number of quartets – but only one quartet per SiteID.

*The SOIK and OSUK are sensitive data, and shall be stored in the SBIO's protected storage.*

The quartet are provided and managed by a Site Management Server. The SBIO requesting a new quartet contacts the corresponding Site Management Server, providing its ObjectID and maybe its user's credentials.

If the Site Management Server accepts the request, the Server computes the SBIO's OSUK and provides a valid quartet in return.

*The Site's Management Server and how it communicates with the SBIO are out of the scope of this document, and under the responsibility of the implementer. The communication shall be secured and the user's credentials carefully handled to prevent any security issue.*

The CRC32 of (SiteID .. UserID) could be processed once for all when the data are loaded and stored with the configuration quartet. It could even be computed by the Site's Management Server.

### 4.3.3. Transaction handling

When the SBIR connects to the SBIO and starts a transaction, the SBIO shall:

1. Receive the 8-byte **SBIR's Challenge**, generate an 8-byte **SBIO's Challenge**, and transmit it in response,
2. Assemble the 16-byte value **Challenges = SBIR's Challenge .. SBIO's Challenge**,
3. Receive the **SiteID** requested by the SBIO, find the corresponding quartet in its configuration data,
4. Assemble the 32-byte value **ObjectID .. SiteID .. UserID .. CRC (SiteID .. UserID)**,
5. XOR both 16-byte halves with **Challenges**,
6. Cipher the 16-byte first half with **SOIK** (AES ECB encrypt),
7. Cipher the 16-byte second half with **OSUK** (AES ECB encrypt),
8. Transmit this 32-byte Secure cryptogram to the SBIR.

### 4.3.4. Exception: unknown Site

If the SiteID requested by the SBIR is unknown from the SBIO, the procedure changes at step 4: the SBIO assemble a 32-byte random value, and transmits it instead of the 32-byte Secure cryptogram.

A SBIR actually belonging to the Site will understand that the SBIO doesn't have any suitable data because its verifications will fails (SiteID, CRC).

A rogue SBIR is not able to distinguish between SBIO returning valid data and a SBIR returning a random value.

# 5. NFC implementation

# 6. BLE implementation

## 6.1. Basis

The SBIR is configured as a BLE Peripheral. It broadcast regularly its advertising data, which allows a BLE Central – i.e., the SBIO – to find it.

The SBIR is a GATT server. The SBIO connects to the SBIR and is able to read or write the characteristics exposed through its GATT.

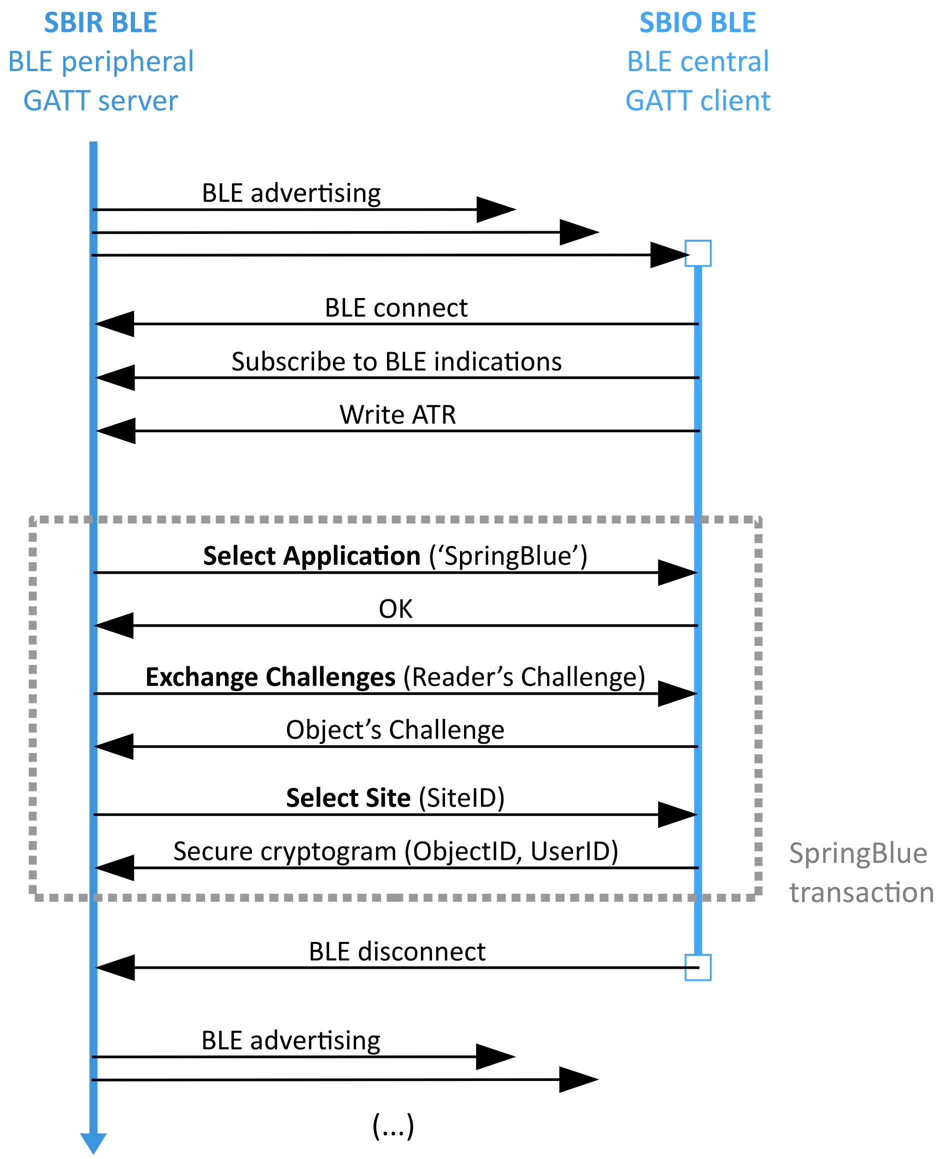

**Illustration 3: BLE implementation of the SpringBlue transaction**

## 6.2. Mapping of commands/responses into BLE read/write frames

Technically speaking, the communication master/slave roles is inverted between BLE and NFC: when the SBIR wants to send a command to the SBIO, the SBIR only notifies the SBIO that a new command is available, and its up to the SBIO to read the command. BLE indications are used to do so. In Illustration 3, every left-to-right arrow (**command SBIR to SBIO**) inside the “SpringBlue transaction” block is actually a 3-step sequence:

- SBIR sends an **indication**,
- SBIO sends a **read request**,
- SBIR sends a **read response**, containing the command buffer.

The SBIO sends back its response by writing into the SBIR. Again in Illustration 3, every right-to-left arrow (**response SBIO to SBIR**) inside the “SpringBlue transaction” block is actually a 2-step sequence:

- SBIO sends a **write request**, containing the response buffer,
- SBIR sends an **aknowledge**.

The communication takes place in unpaired, unbound mode. The security is implemented at application level, not at communication level.

## 6.3. Advertisement data of the SBIR

### 6.3.1. Advertisement frame

| Descriptor #1 | | | Descriptor #2 | | |
|---|---|---|---|---|---|
| **Len** | **Type** | **Data** | **Len** | **Type** | **Data** |
| $_{h}$02 | $_{h}$01 | $_{h}$05 | $_{h}$02 | $_{h}$0A | $_{h}$XX |
| | **Flags** Record | - LE Limited Discoverable Mode<br>- No BR/EDR (BLE only) | | **Tx Power** Record | Tx Power value |

| Descriptor #3 | | |
|---|---|---|
| **Len** | **Type** | **Data** |
| $_{h}$11 | $_{h}$16 | $_{h}$93 F5 A4 62 15 6F 41 B8<br>B8 18 58 BB D3 6F BD CD |
| | **Incomplete list of 128-bit Service Class UUIDs** Record | UUID of the **SpringBlue ID** Service |

### 6.3.2. Scan response frame

| Descriptor #1 | | |
|---|---|---|
| **Len** | **Type** | **Data** |
| ${}_h$12 | ${}_h$09<br>**Local Name** Record | *`SpringBlue XXXXXX`*<br>Advertised name where XXXXXX are the MAC address Least significant bytes. |

## 6.4. GATT profile of the SBIR

### 6.4.1. Standard services and characteristics

| *UUID* | *Mnemonic* | *Description* | *Access* |
|---|---|---|---|
| ***Generic Attribute*** | | | |
| 1801 | org.bluetooth.service.generic_attribute | | |
| 2A05 | org.bluetooth.characteristic.gatt.service_changed | Notifies the BLE central that the GATT should be read again | Read, Indicate |
| ***Generic Access Profile*** | | | |
| 1800 | org.bluetooth.service.generic_access | | |
| 2A00 | org.bluetooth.characteristic.gap.device_name | The name of the SBIR: “SPRINGBLUE” | Read |
| ***Device Information*** | | | |
| 180A | org.bluetooth.service.device_information | | |
| 2A29 | org.bluetooth.characteristic.manufacturer_name_string | “SpringCard” | Read |
| 2A24 | | | Read |

| *UUID* | *Mnemonic* | *Description* | *Access* |
|---|---|---|---|
| | `org.bluetooth.characteristic.model_number_string` | Depend on the actual SBIR product | |
| `2A25` | `org.bluetooth.characteristic.serial_number_string` | The BT_ADDR, in hex | Read |
| `2A28` | `org.bluetooth.characteristic.firmware_revision_string` | Version of the SBIR's firmware (currently "1.00") | Read |

| *UUID* | *Mnemonic* | *Description* | *Access* |
|---|---|---|---|
| *Device information (cont.)* | | | |
| `2A05` | `org.bluetooth.characteristic.software_revision_string` | Version of the SBIR implementation (currently "1.00") | Read |
| ***Tx Power*** | | | |
| `1804` | `org.bluetooth.service.tx_power` | | |
| `2A07` | `org.bluetooth.characteristic.tx_power_level` | The (estimated) transmit power level in dBm, and the level ranges from -100 dBm to +20 dBm, with a resolution of 1 dBm. | Read |

### 6.4.2. SpringBlue service and characteristic

| *UUID* | *Mnemonic* | *Description* | *Access* |
|---|---|---|---|
| **SpringBlue Service** | | | |
| `93F5A462-165F-41B8-B818-58BBD36FBDCD` | | | |
| `9C38A319-F06F-4DD3-AEE4-42747A7307E` | | **SpringBlue APDU Exchange characteristic**<br>Single characteristic used for bi-directional communication. The SBIO shall register to receive the indications on this characteristic. | Read, Write, Indicate |

## 6.5. Format of the SpringBlue APDU Exchange characteristic

# 7. Test vectors

| | |
|---|---|
| SiteID | 00 00 00 01 |
| SOIK | A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF |
| MSUK | B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF |

## 7.1. Object 1

| | |
|---|---|
| ObjectID | 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F |
| OSUK | |
| UserID | 01 02 03 04 05 06 07 08 |
| SiteID .. UserID .. CRC | 00 00 00 01 01 02 03 04 05 06 07 08 xx xx xx xx |

### 7.1.1. Transaction 1

| | |
|---|---|
| SBIR’s Challenge | 00 00 00 00 00 00 00 00 |
| SBIO’s Challenge | 00 00 00 00 00 00 00 00 |
| Challenges | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenges | |
| Ciphered cryptogram | |

### 7.1.2. Transaction 2

| | |
|---|---|
| SBIR's Challenge | C0 C1 C2 C3 C4 C5 C6 C7 |
| SBIO's Challenge | C8 C9 CA CB CC CD CE CF |
| Challenges | C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenges | |
| Ciphered cryptogram | |

### 7.1.3. Transaction 3

| | |
|---|---|
| SBIR's Challenge | C8 C9 CA CB CC CD CE CF |
| SBIO's Challenge | C0 C1 C2 C3 C4 C5 C6 C7 |
| Challenges | C8 C9 CA CB CC CD CE CF C0 C1 C2 C3 C4 C5 C6 C7 |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenges | |
| Ciphered cryptogram | |

## 7.2. Object 2

| | |
|---|---|
| ObjectID | E0 E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF |
| OSUK | |
| UserID | F0 F1 F2 F3 F4 F5 F6 F7 F8 |
| SiteID .. UserID .. CRC | 00 00 00 01 F1 F2 F3 F4 F5 F6 F7 F8 xx xx xx xx |

### 7.2.1. Transaction 1

| | |
|---|---|
| SBIR's Challenge | 00 00 00 00 00 00 00 00 |
| SBIO's Challenge | 00 00 00 00 00 00 00 00 |
| Challenges | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenges | |
| Ciphered cryptogram | |

### 7.2.2. Transaction 2

| | |
|---|---|
| SBIR's Challenge | C0 C1 C2 C3 C4 C5 C6 C7 |
| SBIO's Challenge | C8 C9 CA CB CC CD CE CF |
| Challenges | C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenge | |
| Ciphered cryptogram | |

### 7.2.3. Transaction 3

| | |
|---|---|
| SBIR's Challenge | C8 C9 CA CB CC CD CE CF |
| SBIO's Challenge | C0 C1 C2 C3 C4 C5 C6 C7 |
| Challenges | C8 C9 CA CB CC CD CE CF C0 C1 C2 C3 C4 C5 C6 C7 |
| ObjectID XOR Challenges | |
| (SiteID .. UserID .. CRC) XOR Challenges | |
| Ciphered cryptogram | |

## 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 SPRINGCARD 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 SPRINGCARD 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. SPRINGCARD reserves the right to change the information at any time without notice.

SPRINGCARD doesn't warrant any results derived from the use of the products described in this document. SPRINGCARD 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. SPRINGCARD customers using or selling these products for use in such applications do so on their own risk and agree to fully indemnify SPRINGCARD for any damages resulting from such improper use or sale.

## Copyright notice

## Editor's information

**SPRINGCARD 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