Easy and Safe Pairing for Bluetooth Smart

M. Meli, O. Rion, L. Widmer
Zurich University of Applied Sciences
Institute of Embedded Systems
Winterthur, Switzerland
Marcel.Meli@zhaw.ch

***Abstract*— The configuration of Bluetooth Smart devices in a secure way is a challenging task. Even more so for early versions of this wireless protocol. The use of NFC can facilitate the task, but requires devices and phones that support the NFC standard. This adds costs, both to the smartphone and to the Bluetooth Smart device. Additionally, many smartphones on the market do not (openly) implement NFC. This further limits the number of users that can use that short range RFID standard to configure wireless systems. In this work, we address these issues. We suggest that easy configuration with nearly all smartphones is possible, by using the screen of smartphones to generate an optical code. That signal is detected using a low cost photosensitive element on the side of the Bluetooth Smart device. The information thus generated can be used to generate a key that helps secure the wireless communication between the 2 parties.**

Keywords—Pairing; Smartphone; Wireless; User friendly; security; Bluetooth Smart; NFC

I. INTRODUCTION AND PROBLEM STATEMENT

The Internet of Things revolution foresees billions of devices that communicate together to enable services. In many cases, part or all of the communication will be wireless. Bluetooth Smart will probably, be one of the WPAN (Wireless Personal Area Wireless Network) systems widely used. The sheer number and various qualifications of people that will somehow be involved in the setting up and/or use of those devices indicates that steps have to be taken to ensure that the setting up of systems is user friendly. In parallel, there is no doubt that the nature of some of the services that will be implemented calls for guarding against illegal acquisition and misuse of data.

One of the important challenges in the proliferation of wireless sensors is the pairing of devices that should communicate together [1,2]. We use pairing in a general way here, meaning the initial exchange of information between communicating parties such as to allow the applications to send the proper information to the proper parties. This also includes securing the exchange of application information.

Millions of devices have been shipped since the introduction of Bluetooth Smart (aka BLE, BTLE) in 2010. Many more will ship. The availability of that standard in smartphones has turned it into one of the most important and popular wireless technologies. Several consumer applications have already embraced BLE. There are also applications in where privacy and security are very important (medical devices). It is obvious that the vast majority of users will be people without engineering background. A key to the success is therefore hiding the complexity of the product. That also means making it easy to set up systems and securing communications.

Many applications and chipsets currently on the market implement the early versions of Bluetooth Smart (4.0, 4.1). It is well documented that mechanisms to secure communication based on those early versions are inappropriate [8,9,10]. The reader is advised to consult the references. One way of allowing safe exchange of initial data for security and configuration of the applications is the use of OOB (Out Of Band). Near Field Communication (NFC) is a good protocol for OOB pairing. A Bluetooth Smart sensor is equipped with an NFC tag (or an NFC interface). A smartphone with NFC features will then be able to write/read information in/from the sensor’s NFC tag. Since NFC communication is typically only possible when devices are very close to each other (few centimeters), it is difficult for a third party to sniff that information. In this way, several bytes of vital information can be exchanged by physically bringing smartphone and sensor together (Touch and Pair [2]). The devices can then be taken to their foreseen utilization position from where they will communicate in a secure way. This solution has 2 important disadvantages:

- The use of NFC is possible only if the sensor has an NFC interface (or an NFC tag). The vast majority of Bluetooth Smart solutions do not include NFC. Recently, firms such as Nordic and Toshiba have introduced Bluetooth Smart solutions that also integrate an NFC interface or tag [6,7]. There have also been efforts to add a restricted set of NFC features on Bluetooth Smart chipsets by integrating the needed functionality in the BLE processor firmware [4]. These solutions add non-negligible costs to a device that is price sensitive.

- The smartphone used must be NFC-capable. In order to use that interface, it is obvious that an appropriate smartphone is needed. Many smartphones do not feature NFC, which means that millions of users cannot use OOB pairing based on NFC.

Alternatives to NFC have been suggested. Some are listed in reference [1], including a simple and low-cost method based on optical communication that could easily be used for wireless sensors. The method suggested takes advantage of IrDA components to transfer data from the pairing device to other nodes. In [3], the screen of a computer system was used to update parameters in a simple embedded system application. The optical information sent by the screen was decoded using low-cost elements placed on the smart Torch. It was shown that key parameters on the fob (torch) could easily be updated from a PC or other embedded systems equipped with a screen. Recently, VLC methods have been used to transfer data from a smartphone to another device. Some of the methods require the use of cameras and image algorithms [11,12 and similar papers] to extract information from pictures. This is definitely too complex and expensive for a simple BLE tag. [13] suggests the use of a set of color sensors to decode information from a smartphone. This is a simpler system than those using cameras, but it is still too complex for the simple pairing of BLE tags. The use of different zones on the smartphone to covey the information also takes away some of the simplicity.

In this work, we suggest the use of a simpler system that only requires a detector on the tag side. We demonstrate it practically. It seems to us that in the context of low-cost BLE tags, this is a solution that is low-cost, easy to implement and simple to use.

II. MOTIVATION

Our motivation is summed up below:

- Simplify the use of wireless systems, especially Bluetooth Smart by providing a configuration method that is easy, intuitive, low-cost and available to most users.
- The configuration method should allow the secure exchange of encryption data and other parameters within a reasonable time.

III. THE CONCEPT

In order to address the issues mentioned earlier, we looked for a method that can be implemented on (almost) all smartphones, yet allow an Out Of Band exchange of data. Smartphones normally have a display unit that is used to communicate information to the user. That display can be programmed to output signals of different illuminations [3], thus optically coding information from the smartphone. This information is captured by a photosensor built in the Bluetooth Smart tag facing (placed against) the screen of the smartphone. The illumination pattern generated can be decoded with simple firmware on the sensor side.

Using the display to communicate information to the sensor presents the following advantages:

- The method can be implemented on nearly all smartphones (better coverage than NFC). It can also be implemented on PCs, tablets, and other devices that sport an appropriate display.
- A pattern of the appropriate illumination can easily be programmed and run on the smartphone (app)
- It is difficult to sniff the optical communication. The use of a small portion of the screen where the tag can be placed and the reduction of the illumination help further restrict a non-authorized access to the optical communication.
- The Bluetooth Smart device only needs a simple photodiode and eventually an amplifier (and appropriate circuitry) to receive the optical signal. Firmware for decoding the pattern is minimal and simple.
- The whole screen or part of it can be used, allowing another application to use the rest of the screen.
- The interaction with a user is simple. It could consist in placing the smartphone and the wireless sensor near one another, starting the data communication by making the correct menu selection on the pairing app.

There are also some potential weak points in this method.

- It is not standardized, meaning that solutions will be proprietary and an appropriate app is needed for each Bluetooth Smart sensor.
- The communication can be complicated by the presence of too much ambient light (this can be mitigated by a proper protocol design for the optical communication).
- The communication speed is low since the screen update rate is low. But there is not much data to transfer for an initial key.

IV. PROOF OF CONCEPT AND RESULTS

In order to verify our concept, we implemented the needed hardware parts and wrote the appropriate app and firmware. In a first run, a Bluetooth Smart kit with a Nordic device [7] was used as sensor. In a second run, a tag with a BLE device from Renesas [14] was used. A smartphone with an Android OS and Bluetooth features was used as host. The system was used to generate a key, send it optically to the Bluetooth smart tag. It was quite easy to add a sensor on the tags to detect light variations and pass the information to the microcontroller. Upon reading the data, an encryption key was generated and used to wirelessly send a message (through Bluetooth Smart advertising channels). That message could obviously only be decrypted by a smartphone in possession of the correct key.

For the proof of concept, we used a short encryption code that was then extended on the sensor and on the smartphone to yield the 128-bit key needed for the AES encryption. Mechanically, the tag and the smartphone were in contact. This makes it intuitive for a user (just put the tag on the area of the smartphone that is marked for blinking) and at the same time reduces the influence of external light source. We call this Place and Pair. The data rate of the optical communication is low (depends on the refresh rate of the phone). 100-200 bits are enough for an initial pairing. After that, the process can continue using an encrypted link if there is a need to exchange more data. It means that after placing the BLE tag on the screen and starting the pairing application, the user needs not wait more than a few seconds. Since a BLE link is available, we did not try at this stage to implement a new feedback channel. There are several ways of avoiding unwanted pairing/data transfer activities. A mechanical lid can be used that will cover the light sensor. A switch can be used to start the procedure. One could also use a special pattern at the start, on lock the pairing electronically after it has been done.

Some of the results are shown below, on a larger format (to help the understanding). The results are commented (Figures 4-5). The reader is kindly asked to refer to those parts.

V. CONCLUSION

In this work, we have shown that important restrictions presently encountered in pairing devices for wireless communications can be overcome with the use of simple electronics and associated firmware and app. The solution is at least as intuitive as NFC. It can be implemented on most smartphones, tablets, PCs, etc. The CPU resources needed are minimal, meaning that most Bluetooth Smart embedded systems can cope with these requirements. Draw-backs such as the speed of the communication using the smartphone display are not an important issue because the amount of data to exchange is small. The influence of external light can be mechanically kept minimal, further simplifying the system.

Future works will concentrate on making long term measurements to quantify the system. We will also make verification on different smartphones. If needed, improvements will be added. A simple bidirectional communication where LED is used to send information to the smartphone (LED to smartphone camera) is also under development. Since LEDs can be used so sense as well one can use 1 LED for both directions (sequentially).

The system can clearly also be used to transfer a small amount of data as done in [3] for many other applications (with or without wireless). It could be time setting, data from a smartphone to a watch, setting information for wearables, … etc.

ACKNOWLEDGMENTS

Bluetooth, Bluetooth Smart, Bluetooth Smart Ready are registered trademarks of Bluetooth SIG, Inc. All other registered trademarks or trademarks are the property of their respective owners.

REFERENCES

[1] Low cost solutions to pairing issues in IEEE 802.15 .4 networks
M. Meli; M. Gysel; M. Wuerms;
1st European ZigBee Developer's Conference, Munich 2007

[2] A Touch and Pair system for battery-free 802.15.4/ZigBee Home Automation networks
D.Condrau; L.Zimmermann; M.Gysel; M. Würms; M.Meli;
3rd European Developer’s ZigBee Conference, Munich, June 2009

[3] Ecological intelligent torch; Jonas Dünki, Raphael Josef
Bachelor project 2012, ZHAW (supervised by M. Meli)

[4] Low energy and highly secured wired to wireless converter
P. Burger, S. Trowbridge ; Bachelor project 2014, ZHAW (supervised by M. Meli)

[5] Bluetooth Secure Simple Pairing Using NFC; Application Document, NFC Forum, NFCForum-AD-BTSSP_1_1 2014-01-09

[6] TC35670; Bluetooth Single-Chip Controller for Bluetooth Low Energy (4.1) + NFC Tag; Toshiba
http://toshiba.semicon-storage.com/eu/product/wireless-communication/bluetooth/TC35670.html

[7] nRF52 Series SoC (Bluetooth Smart transceiver integrating NFC)
https://www.nordicsemi.com/Products/nRF52-Series-SoC

[8] Bluetooth Smart: The Good, the Bad, the Ugly, and the Fix!
Black Hat USA 2013; https://www.youtube.com/watch?v=SoH11fi-FcA

[9] Bluetooth: With Low Energy Comes Low Security; Mike Ryan, iSEC Partners; https://www.usenix.org/conference/woot13/workshop-program/presentation/ryan

[10] Security Considerations For Bluetooth Smart Devices
Ravikiran HV, PathPartner Technology Pvt.Ltd.
http://www.design-reuse.com/articles/39779/security-considerations-for-bluetooth-smart-devices.html

[11] Novel detection technique for smartphone to smartphone visible light communications. R. Boubezari; H. Le Minh; Z. Ghassemlooy; A. Bouridane; 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)

[12] SBVLC: Secure Barcode-Based Visible Light Communication for Smartphones; Bingsheng Zhang; Kui Ren; Guoliang Xing; Xinwen Fu; Cong Wang; IEEE Transactions on Mobile Computing Year: 2016

[13] NECAS: Near field communication system for smartphones based on visible light; Jianwei Niu ; Wenfang Song ; Chuang Liu ; Lei Shu ; Canfeng Chen; Publication Year: 2014, Page(s):2426 – 2431

[14] Renesas Bluetooth Smart RL78/G1D Evaluation Board
https://www.renesas.com/en-eu/products/software-tools/boards-and-kits/evaluation-demo-solution-boards/rtk0en0001d01001bz.html.html

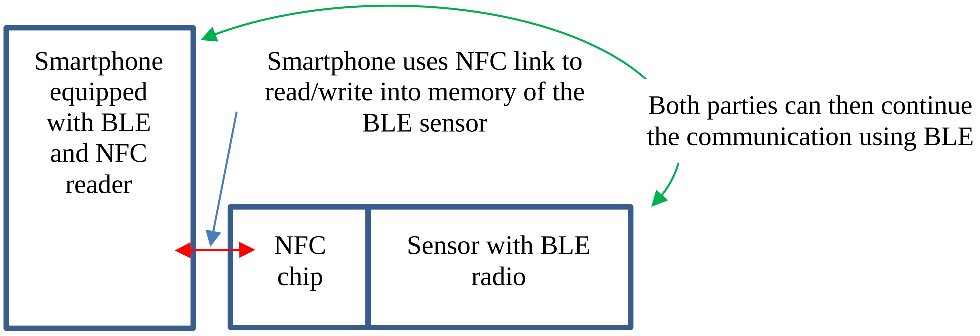

Smartphone equipped with BLE and NFC reader
Smartphone uses NFC link to read/write into memory of the BLE sensor
Both parties can then continue the communication using BLE
NFC chip
Sensor with BLE radio

Fig. 1 Basic block diagram of a system using NFC for pairing
Both parties can communicate when near each other via NFC. A smartphone equipped with NFC can then write a code in the NFC tag of the sensor. This code that is known by both parties will be used to encrypt the BLE communication.

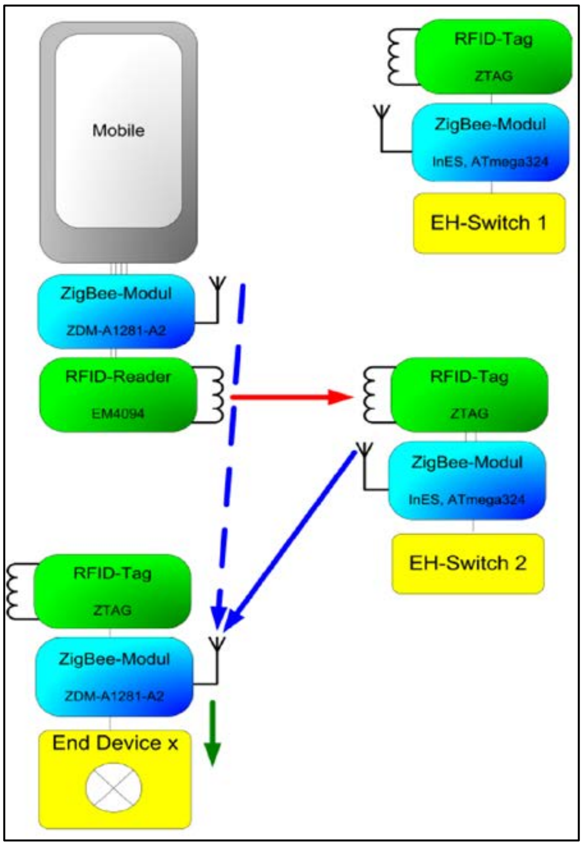


Fig.2 System using dual ported RFID tags for pairing and association of ZigBee modules (especially battery free switches), as proposed in reference [2]. When the switch is pressed, energy is delivered and used to send a message to switch a load on/off. Such switches cannot scan or associate because they do not have energy. At installation, information about the proper channels, end device, AES code is written in their memory using an RFID reader. It is obvious that the system can also be used for other wireless systems.

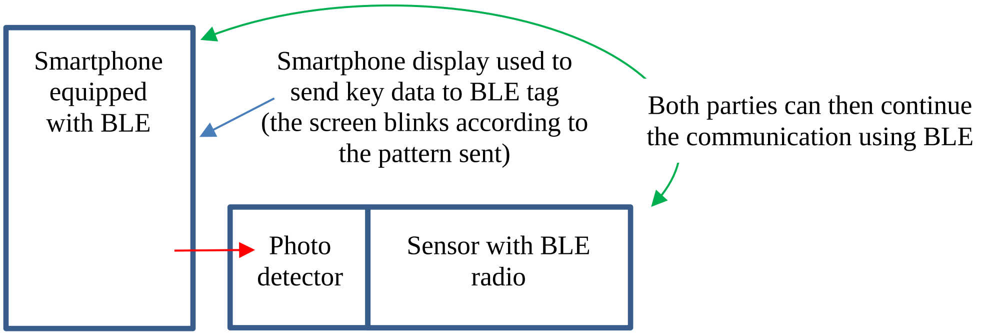
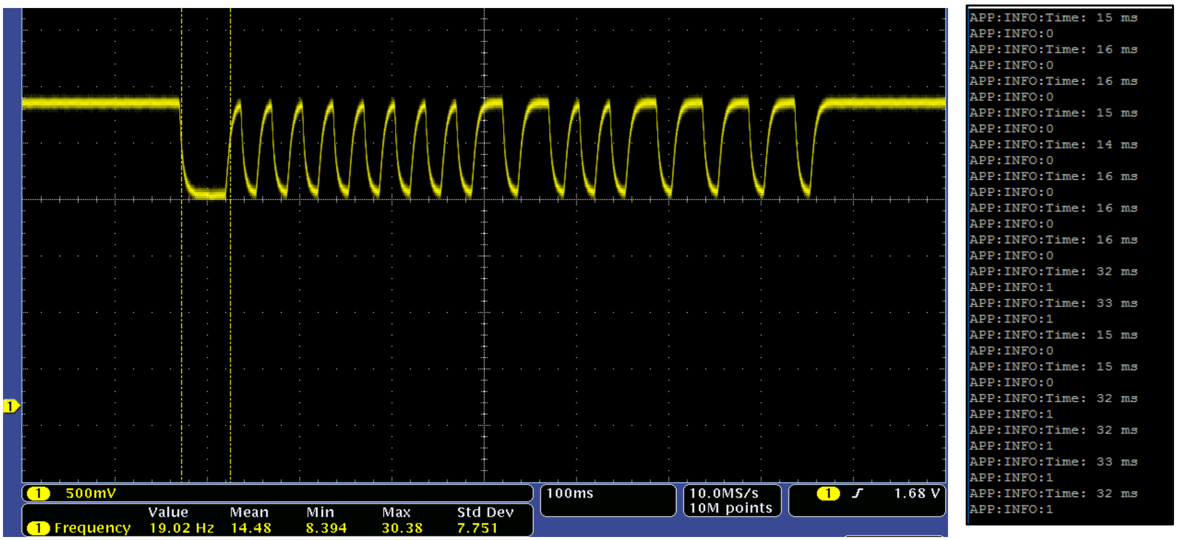

Smartphone equipped with BLE
Smartphone display used to send key data to BLE tag (the screen blinks according to the pattern sent)
Both parties can then continue the communication using BLE
Photo detector
Sensor with BLE radio

Fig.3 Basic diagram of the proposed pairing method. As in the case of NFC, the BLE tag and the smartphone are very near. This insures a proper communication while making it difficult for a third party to “listen”. The key that is exchanged optically is a random pattern generated by the smartphone and sent to the tag. That pattern is used to feed the AES block for encryption of the communication. Verification and retries can be built-in if necessary. Once the procedure is completed, a secure exchange is possible.


1
1 500mV
100ms
10.0MS/s
10M points
1 1.68 V
Value Mean Min Max Std Dev
1 Frequency 19.02 Hz 14.48 8.394 30.38 7.751

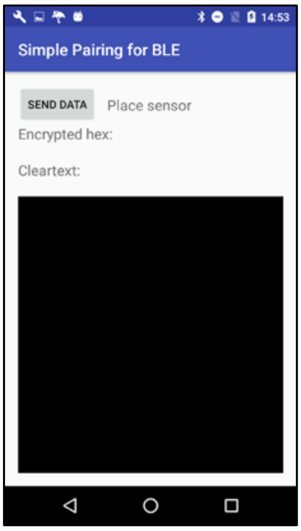
```
APP:INFO:Time:  15 ms
APP:INFO:0
APP:INFO:Time:  16 ms
APP:INFO:0
APP:INFO:Time:  16 ms
APP:INFO:0
APP:INFO:Time:  15 ms
APP:INFO:0
APP:INFO:Time:  14 ms
APP:INFO:0
APP:INFO:Time:  16 ms
APP:INFO:0
APP:INFO:Time:  16 ms
APP:INFO:0
APP:INFO:Time:  16 ms
APP:INFO:0
APP:INFO:Time:  32 ms
APP:INFO:1
APP:INFO:Time:  33 ms
APP:INFO:1
APP:INFO:Time:  15 ms
APP:INFO:0
APP:INFO:Time:  15 ms
APP:INFO:0
APP:INFO:Time:  32 ms
APP:INFO:1
APP:INFO:Time:  32 ms
APP:INFO:1
APP:INFO:Time:  33 ms
APP:INFO:1
APP:INFO:Time:  32 ms
APP:INFO:1
```

Fig.4 Oscilloscope picture of the signal at the optical receiver when the screen of the smartphone blinks. A “0” is coded by using a short pulse. A “1” is coded with a long pulse

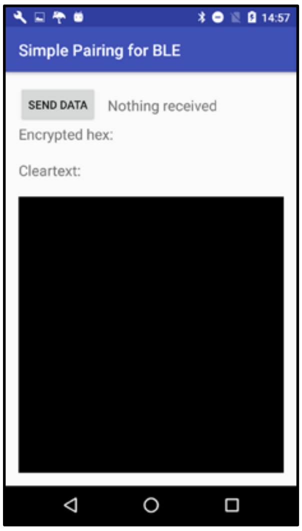
A single byte was used for the proof of concept, with necessary guard information (not all shown here) The rest of the key was filled in with FF for the sake of simple illustration.

Parts used for the proof of concept

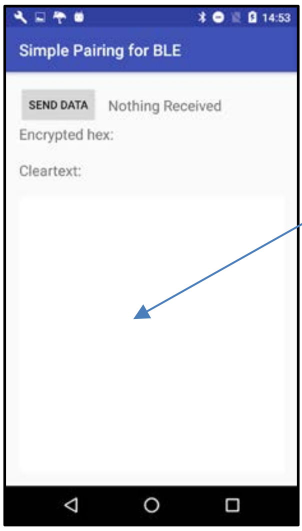
Device	Details
Smartphone	moto g 3 running Android 6.0
BLE device1	Nordic Nrf52 dk
BLE device2	Renesas RL78/G1D
Sensor	Vishay ambient light sensor 751-1055-1-ND



Place sensor
Encrypted hex:
Cleartext:



Nothing received
Encrypted hex:
Cleartext:



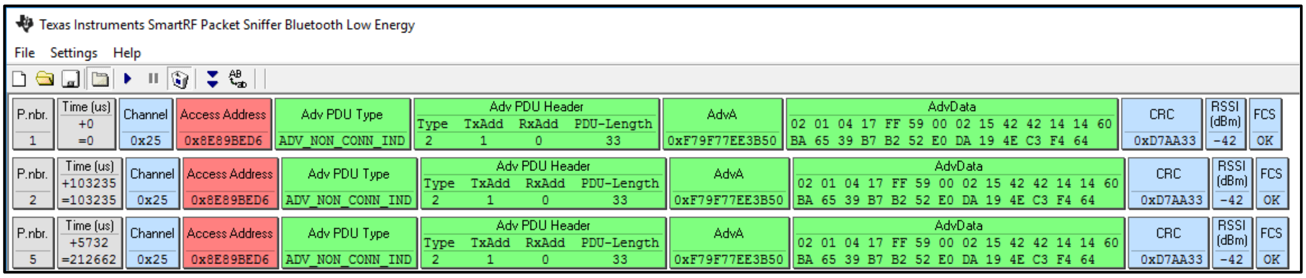
Nothing Received
Encrypted hex:
Cleartext:

The smartphone and the sensor are brought in «near contact»

The user simply places the tag here, with the light sensor facing the screen

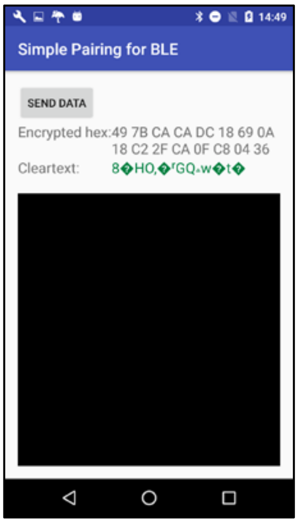
Pairing is started from the smartphone

The illumination of the screen of the smartphone changes in function of the data that is sent.

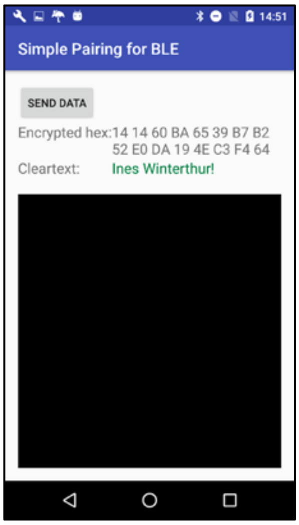


Texas Instruments SmartRF Packet Sniffer Bluetooth Low Energy
File
Settings
Help

P.nbr.	Time (us)	Channel	Access Address	Adv PDU Type	Adv PDU Header Type	TxAdd	RxAdd	PDU-Length	AdvA	AdvData	CRC	RSSI (dBm)	FCS
1	+0 =0	0x25	0x8E89BED6	ADV_NON_CONN_IND	2	1	0	33	0xF79F77EE3B50	02 01 04 17 FF 59 00 02 15 42 42 14 14 60 BA 65 39 B7 B2 52 E0 DA 19 4E C3 F4 64	0xD7AA33	-42	OK
2	+103235 =103235	0x25	0x8E89BED6	ADV_NON_CONN_IND	2	1	0	33	0xF79F77EE3B50	02 01 04 17 FF 59 00 02 15 42 42 14 14 60 BA 65 39 B7 B2 52 E0 DA 19 4E C3 F4 64	0xD7AA33	-42	OK
5	+5732 =212662	0x25	0x8E89BED6	ADV_NON_CONN_IND	2	1	0	33	0xF79F77EE3B50	02 01 04 17 FF 59 00 02 15 42 42 14 14 60 BA 65 39 B7 B2 52 E0 DA 19 4E C3 F4 64	0xD7AA33	-42	OK



Encrypted hex:49 7B CA CA DC 18 69 0A 18 C2 2F CA 0F C8 04 36
Cleartext: 8�HO,�ʳGQ˳w�t�



Encrypted hex:14 14 60 BA 65 39 B7 B2 52 E0 DA 19 4E C3 F4 64
Cleartext: Ines Winterthur!

The data sent by the smartphone is a random number that is used to generate the 128-bit AES key.
The sensor generates that key and uses it to encrypt a message (In our example: **Ines Winterthur!**).
The encrypted message is broadcasted (ADV frames) using BLE (sniffer output).
A user without the key will not be able to decrypt the message.
The smartphone that was used for pairing knows that key and will decrypt the message correctly.

Fig.5 Illustration of a sequence that demonstrates the concept.

Settings as below.

Encryption, Keys, Plaintext and Ciphertext

Encryption:	AES 128 bit ECB
Key (hex):	cf ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff -> Only 'cf' is sent by the phone's display
Plaintext:	Ines Winterthur!
Plaintext (hex):	49 6E 65 73 20 57 69 6E 74 65 72 74 68 75 72 21
Ciphertext (hex):	14 14 60 BA 65 39 B7 B2 52 E0 DA 19 4E C3 F4 64