Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Master Projekt 1

Autor: Heiner Perrey

The Concept of Bluetooth Low Energy with Merkle's Puzzle

Date: May 20, 2012

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Contents

1 Introduction

Note

Part of this work has been published at WiSec '11 and in ACM SIGMOBILE [9]. Some sections have been changed due to new research and conclusions.

The work at hand is a conceptual study attempting to provide a supplementary security feature in Bluetooth Low Energy (BLE). BLE, prior known as *Wibree* [17], is a fairly new technology with the goal to connect energy restricted devices or devices with asymmetric capabilities, e.g. a smartphone and a sensor node with one another [13]. Although BLE is based on the classic Bluetooth (BT) standard, both are not directly compatible. The main difference is the reduced energy consumption of BLE which comes at the cost of a lower data transmission rate and a lower security level during pairing.

BT uses Secure Simple Pairing (SSP) to securely couple two devices. SSP provides protection against passive eavesdropping by employing Elliptic Curve Diffie-Hellman (ECDH) for key exchange. As ECDH is not included in the latest BLE specification, it is vulnerable to passive attacks. Even though later versions of the BLE specification will implement ECDH, it may not be operable on extremely restricted devices, like bio-sensors or other types of reduced functioning devices [1, 2].

Merkle's Puzzle (MP) is one of the first protocols which allows two or more parties to securely agree upon a shared key solely using an insecure channel [5]. The most important feature for this work is the adjustability of the workload with respect to the capabilities of the weaker device. Thus, if the devices have significantly asymmetric capabilities, the workload may be shifted nearly completely to the stronger device. This holds the precondition of operating with a fully functional device (FFD) and one or more reduced functional devices (RFD) such as for W-BANs [1].

In this work an alternative light-weight key agreement protocol for BLE based on MP is proposed. Although the security of MP in practice is only temporary, it is very promising for extremely limited devices which otherwise would be completely insecure or for which a periodical key refreshment is applicable. Since the devices are reduced in their function, it is important to keep the additional expenses at a minimum. Therefore a security model using MP directly placed into the BLE protocol is proposed. In this work practical approaches in the specification to which MP can be added are shown.

2 Bluetooth Low Energy with Merkle's Puzzle

2.1 Merkle's Puzzle

MP is one of the first key agreement protocols introduced in 1974 and published in 1978 [5, 15]. It laid the groundworks for the Diffie-Hellman key agreement protocol [4]. In this work MP for BLE is considered in application scenarios where an FFD wants to securely exchange a key with one (or multiple) RFD as described in [1]. The RFD is not able to run complex protocols such as ECDH due to its low capabilities. Since MP allows to shift nearly the whole workload to the FFD at the benefit of the RFD, both are able to securely agree upon a key. However, in practice the key may only be secure for a roughly defined period of time as described next.

To agree upon a key using MP, the FFD and RFD follow the steps as depicted in figure 2.1. First the FFD creates n puzzles of the form $P_i = E_{k_i}(P_{ID_i}, K_i)$, where P_{ID} is a random and unique puzzle identifier and K a strong key. Each puzzle is encrypted with a function E and a weak key k, which may be as weak as desired. Every k is chosen such that the RFD breaks the encryption within an adequate period of time using brute force. The FFD broadcasts all n puzzles. By randomly choosing a time interval during the broadcast the RFD receives only one out of n puzzles. The RFD then decrypts the received puzzle to obtain the specific P_{ID} and the strong key K. As each puzzle is encrypted with a weak encryption key k, the decryption via a brute-force attack is possible. Finally, the RFD sends back the P_{ID} to the FFD after the broadcast of all puzzles is done. As the FFD can associate the P_{ID} to the right puzzle, the FFD and RFD now share the same strong key K.

An eavesdropper, Eve, may know all puzzles and the P_{ID} sent by the RFD. Since each P_{ID} is chosen randomly and broadcasted in encrypted puzzles, Eve must randomly pick puzzles and break k, until she finds the one with the right P_{ID}. She must break approximately $\frac{n}{2}$ puzzles, therefore requiring a quadratic complexity relative to the linear costs of the FFD and RFD.

With the security parameters n and the size of k, in theory any desired proportion of security and performance is adjustable. However, practically the proportion is limited by the capabilities of the FFD and RFD and time restrictions with respect to the key establishment.

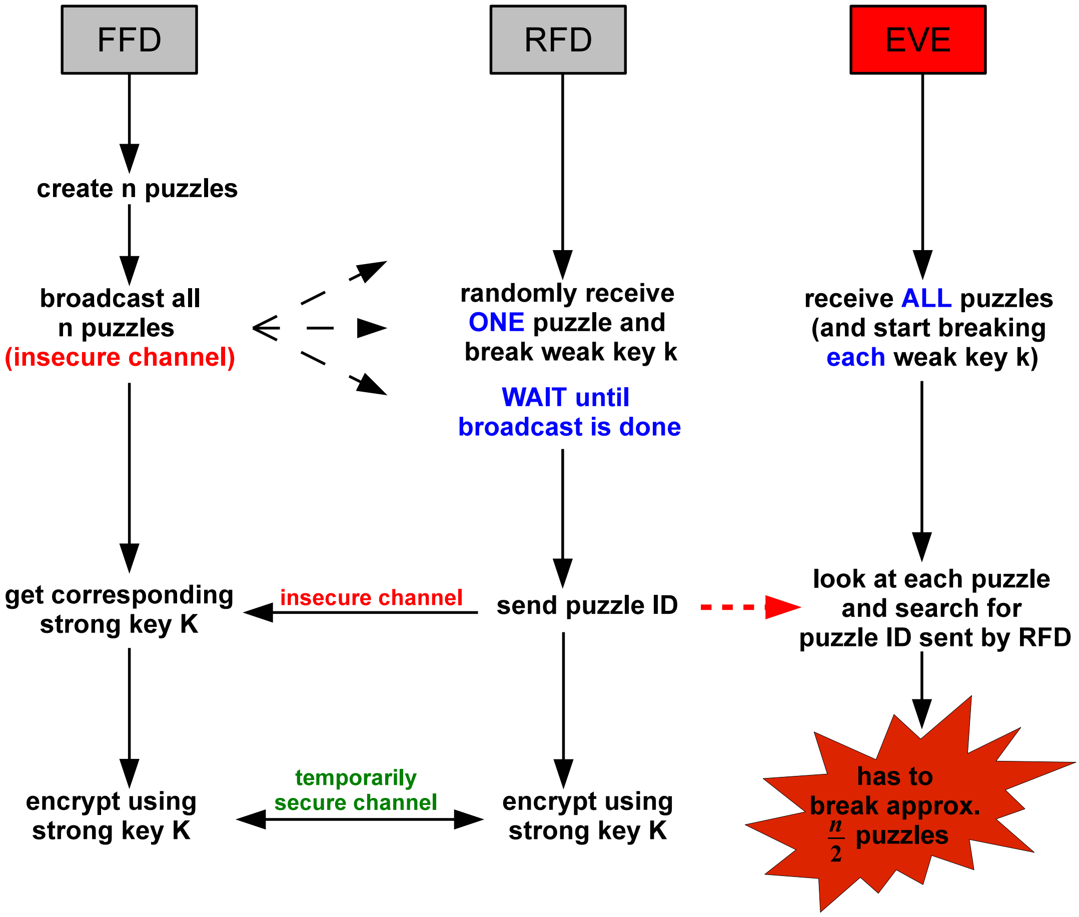

Figure 2.1: Flowchart of Merkle's Puzzle

However, when using Merkle's Puzzle the encrypted data sent right after the key agreement is secure over a longer period of time than the encrypted data sent right before the key expires. Therefore to keep all data equally secure, a *maximum refresh cycle* (MRC) has to be defined. For instance, to keep all data secure for three days, the MRC is set to three days. Not considering the time for key agreement, a potentially *six-day-key* is exchanged. After three days the key has to be refreshed. As the key is secure for about six days, the last data packet sent on the third day is secure for three more days.

2.2 Bluetooth Low Energy Protocol Stack

The goal of this work is to integrate MP into the BLE specification. This section gives a brief overview of the Bluetooth low energy protocol stack as defined in the Bluetooth specification version 4.0 [2]. This document also contains more details on classic Bluetooth [2]. For attacks and security aspects of classic Bluetooth see [3, 16].

Figure 2.2 shows the outline of the BLE stack. Each application is implemented on top of the Generic Access Profile (GAP) or Generic Attribute Profile (GATT). The stack is basically divided into two parts: the host and the controller. The host performs higher level protocols like security features. The controller performs lower level functions like link management or baseband operations on the physical channels [2]. The host contains the rest of the application [14]. The following section gives more details about both the host and the controller part.

2.2.1 Generic Access Profile (GAP)

GAP directly communicates with the application. In GAP the idle and connecting mode procedures are defined. Idle mode procedures include the discovery of Bluetooth devices, while connecting mode procedures manage the connection to a device. GAP also offers higher level security procedures, like the encryption of a connection [2, 12].

2.2.2 Attribute Protocol (ATT)

ATT defines attributes which can be discovered, written to or read from a device. A client and a server role are distinguished. The device receiving an attribute is in the role of the *client*, the device offering the attribute is in the role of the *server* [2].

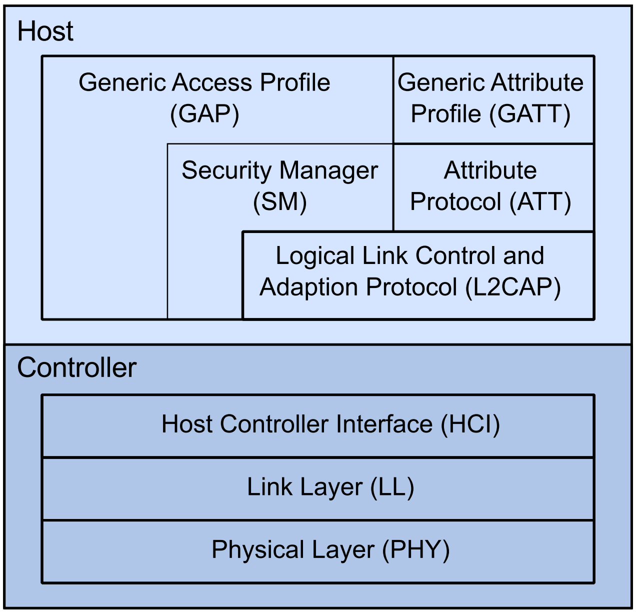
Host
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Security Manager (SM)
Attribute Protocol (ATT)
Logical Link Control and Adaption Protocol (L2CAP)
Controller
Host Controller Interface (HCI)
Link Layer (LL)
Physical Layer (PHY)

Figure 2.2: BLE Protocol Stack [12]

2.2.3 Generic Attribute Profile (GATT)

GATT defines all necessary function for using ATT. GATT is used for the actual data exchange of devices in a connection. The application will use GATT directly instead of using ATT. An example could be a humidity sensor node sending its data to a client. A computer acting as the client may request the specific value of the attribute *humidity*. The sensor node, the server, will then write the value to the client [2, 11].

2.2.4 Security Manager (SM)

SM offers all functionality regarding the security, like pairing, authentication and encryption between devices using BLE. For encryption AES-128 is available. SM also offers a procedure to randomly generate the private address. The current version of the specification does not offer eavesdrop protection during pairing [2].

2.2.5 Logical Link Control and Adaptation Protocol (L2CAP)

The L2CAP is located above the LE controller. It offers the basic functionality for the communication of the lower layer with the upper layer protocols. The fragmentation (and latter recombination) of PDUs before passing them on to the lower layers is done by L2CAP. It also offers protocol/channel multiplexing of data by different protocols [2]. This is done under the assumption that the controller has lower resources, e.g. like buffer capabilities [14].

2.2.6 Host Controller Interface (HCI)

HCI manages the communication between the host and the controller part of the BLE stack [2, 12]. HCI provides an interface to the link layer and the BLE baseband. Hereby, for instance, a USB interface may be placed between the host and the controller [14]; this can be done in a Bluetooth USB dongle.

2.2.7 Link Layer (LL)

LL controls the state of the BLE device. The device may enter the states *standby*, *advertising*, *scanning*, *initiating*, and *connection*. The advertising state allows a BLE device to discover or connect to other devices, or to broadcast user data. Advertisements were first introduced in the BLE part of the Bluetooth specification 4.0. Three channels have been reserved to send advertisements: channel 37, 38, and 39. Hereby each advertisement is sequentially sent on each channel once. The device listening to advertisements needs to be in the scanning state. More details on advertisements are depicted in section 2.3. The initiating state may be entered to initiate a connection and the connection state to actively participate in a connection. The standby state may be entered to safe power consumption, when no messages are exchanged [2, 12].

2.2.8 Physical Layer (PHY)

PHY defines the BLE baseband matter. BLE devices operate on the frequency of 2.4 GHz ISM (Industrial Scientific Medical) band. Like in Bluetooth, BLE uses frequency hopping over 40 radio frequency channels to reduce the probability of collisions [2].

2.3 Enhancing the BLE Specification

This work is based on the idea of building up on the BLE standard and to illustrate any modification necessary for a new version of the specification that natively supports the BLE with MP approach. This section presents the proposed concept and also depicts the requirements to the stack.

Currently pairing is used for the exchange of an encryption key. Prior to the pairing a Link Layer connection has to be created as the key information is sent on a different channel than

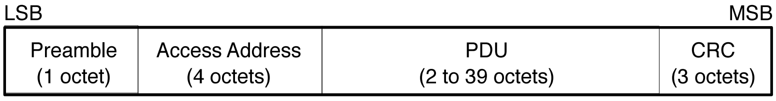

Figure 2.3: *Link Layer packet format [2]*

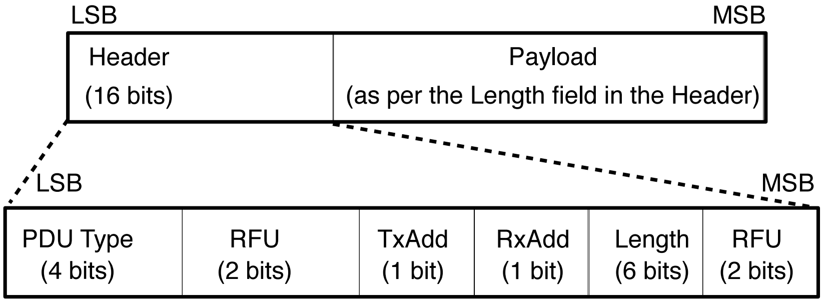

Figure 2.4: Advertising channel PDU and Header [2].

e.g. device discovery information. MP could be considered as a new additional association model, like *Passkey Entry* [2]. Instead of typing a randomly generated number into the other device, the key is generated on the FFD and distributed according to MP. The Security Manager of BLE must therefore be enhanced with the MP logic.

To work closely to the BLE specification the provided security functions should be used. At the moment BLE only provides AES-128 for encryption, which results in too many padding bit as depicted in [8]. Therefore RC5 with a block size of 32 bit is proposed as an additional security function [10]. Furthermore, the BLE controller has to offer the remaining infrastructure to run MP. As described next, this work aims at a connectionless solution, since the puzzles may be sent to many devices concurrently.

In theory an FFD broadcasts puzzles to one or many RFDs. For an efficient distribution, a connectionless approach is suggested, as opposed to the current creation of an LL connection. Only for transmitting the P_{ID} back to the FFD, a unicast message is needed. Hereby a newly introduced way to broadcast user data, called advertisements introduced by Bluetooth 4.0, may provide an easy way to broadcast the Merkle puzzles [2]. With advertisements a

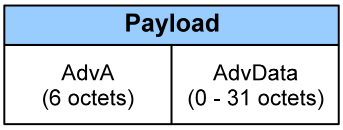

Figure 2.5: *Payload of an ADV_IND PDU [2]*

device may either discover, connect or broadcast user data to other devices (see section 2.2.7). Advertisement packets are sent within Link Layer packets, the format of which is shown in figure 2.3. It consists of a *preamble*, the *access address* (for accessing a connection of a BLE device), the actual (advertising) message or *Protocol Data Unit* (PDU) and a *cyclic redundancy check* (CRC).

Figure 2.4 depicts the basic setup of such an advertising channel PDU. The header is composed of the *PDU Type* field, *TxAdd* and *RxAdd* fields for specific information about the PDU type and some fields reserved for future use (RFU). The payload of the PDU is of variable size. A PDU is part of an LL packet with a maximum size of 376 bit, depending on the size of the PDU payload. There are different PDU types for different events. For a connectable undirected advertising event, for instance, the payload consists of 0–31 byte for the data and 6 byte for the advertiser's address. The use of this type of PDU as a carrier for one puzzle, $P_i = E_{k_i}(P_{ID_i}, K_i)$, would allow to broadcast puzzles of the size up to 248 bit inside the payload of the PDU. Figure 2.5 shows the architecture of such a PDU's payload. The field *AdvA* contains the (MAC) address of the advertiser. The actual data is stored in the *AdvData* field. In this case the Merkle puzzles are placed in this field. In this application scenario of MP with BLE, the FFD is equivalent to a BLE advertiser and the RFD analogue to the advertisement scanner. The following list shows the different types of advertisements and their defined timeout (*TO*) between each data packet:

- ADV_IND: connectable undirected advertising event (*TO* $>=$ 20 ms)
- ADV_DIRECT_IND: connectable directed advertising event (*TO* $>=$ 20 ms)
- ADV_NONCONN_IND: non-connectable undirected advertising event (*TO* $>=$ 100 ms)
- ADV_SCAN_IND: scannable undirected advertising event (*TO* $>=$ 100 ms)

Since this *TO* lowers the performance quite noticeably (as depicted in [8]), a new type *ADV_BLEMP* is necessary to send the puzzles with a shorter or even without a delay. Hereby the effect of potentially more advertisements colliding needs to be examined. However since the advertisements are an unreliable transmission format, not all collisions can be eliminated [2].

The use of advertisements appears to be a promising way to distribute the puzzles without a connection. Since the advertisements are currently not intended to exchange a key, this part of the specification has to be modified according to the additional requirements for MP.

To further increase the performance all three advertisement channels need to be used. For BLE with MP, three different advertisements have to be sent simultaneously on all three advertising channels. However, it has to be guaranteed that none of these alterations conflict with a basic principle of the specification. In this case, a new link layer state in addition to advertising has to be created. However, this may result in major changes and is therefore subject to future research.

2.4 Conclusion

The combination of BLE with MP seems to be a promising way to equip RFD with a secret key for communication with an FFD. Hereby different kind of scenarios are possible. For instance, many RFD may simply send their measured data to the main computer (FFD). Or the FFD could not only function as central device but also be used as a router for the communication between two or more RFD. Hereby it knows all secret keys and can wrap and unwrap the data packets before redirecting them to their target. However, further research like a performance analysis is required to corroborate the BLE with MP approach.

3 Overview and Outlook

BLE is a new energy saving technology, which is especially attractive for extremely restricted devices. Although future versions will implement ECDH and therefore passive eavesdrop protection this may not be applicable for various devices with asymmetric capabilities. When using MP, protection against passive attackers can be integrated even for highly limited devices, which otherwise would not be able to exchange a secret key. The FFD has to bear the computational burden for both, but for the RFD hardly any additional expenses arise.

Section 2.1 introduced the procedure of MP. Section 2.2 gave insights in the BLE protocol stack. Hereby the host and controller parts have been depicted to give the necessary fundamentals for the concept of this work. Section 2.3 gave an overview of how to integrate MP into the BLE specification. The advertisement function of BLE was characterized as an easy way to distribute the puzzles. Hereby an existing part of the specification can be used which, however, still has to be customized. Hereby e.g. the additional security primitive RC5 was mentioned. Also the need for a new advertisement type was emphasized. In this context the defined timeouts between the sending of each advertisement have been addressed.

For future research an implementation which runs with a customized BLE stack and expressive benchmarks is intended. Hereby the sequel work [8] offers a first performance analysis and the work with a practical BLE development kit.

Bibliography

[1] F. Armknecht and D. Westhoff. Using Merkle's Puzzle for Key agreement with Low-end Devices. *IEEE 34th Conference on Local Computer Networks, 2009. LCN 2009.*, pages 858–864, December 2009.

[2] BLUETOOTH SPECIFICATION Version 4.0. Document - Bluetooth SIG, June 2010.

[3] J. Cache, J. Wright, and V. Liu. *Hacking Exposed Wireless, Second Edition.* Hacking Exposed. McGraw-Hill, 2010.

[4] W. Diffie and M. Hellman. New Directions in Cryptography. *Information Theory, IEEE Transactions on*, pages 644–654, November 1976.

[5] R. Merkle. Secure Communications Over Insecure Channels. *Communications of the ACM*, pages 294–299, April 1978.

[6] H. Perrey. Angriffe auf Funknetzwerke. Ausarbeitung Masterkurs „Anwendungen 1“, Hochschule für Angewandte Wissenschaften, Hamburg, February 2011.

[7] H. Perrey. Attacks on Wireless Networks. Ausarbeitung Masterkurs „Anwendungen 2“, Hochschule für Angewandte Wissenschaften, Hamburg, June 2011.

[8] H. Perrey. Performance Analysis of Bluetooth Low Energy with Merkle's Puzzle. Ausarbeitung Masterkurs „Projekt 2“, Hochschule für Angewandte Wissenschaften, Hamburg, February 2012.

[9] H. Perrey, O. Ugus, and D. Westhoff. WiSec' 2011 poster: Security Enhancement for Bluetooth Low Energy with Merkle's Puzzle. *ACM SIGMOBILE: Mobile Computing and Communications Review*, 15(3):45–46, July 2011.

[10] R. Rivest. The RC5 Encryption Algorithm. *Proceedings of the 1994 Leuven Workshop on Fast Software Encryption (Springer)*, pages 86–96, 1995.

[11] B. Tanner and G. Gräni. Sensornetzwerk mit Bluetooth Low Energy. Bachelorarbeit, Zürcher Hochschule für Angewandte Wissenschaften, Winterthur, June 2011.

[12] Bluetooth® Low Energy Software Developer's Guide v1.2. Texas Instruments - Document. http://www.ti.com.cn/cn/lit/ug/swru271b/swru271b.pdf - last checked: 09.03.2012.

[13] Bluetooth.com. webpage, 2012. http://www.bluetooth.com/ - last checked: 02.03.2012.

[14] KnowledgeCenter: More than just replacing cables. webpage. http://developer.bluetooth.org/KnowledgeCenter/TechnologyOverview/ - last checked: 13.05.2012.

[15] Homepage of Ralph Merkle. webpage. http://www.merkle.com/ - last checked: 19.02.2012.

[16] Project Ubertooth. webpage. http://ubertooth.sourceforge.net/ - last checked: 19.02.2012.

[17] Wibree becomes ULP Bluetooth. webpage. http://www.electronicsweekly.com/Articles/2007/06/12/41582/Wibree-becomes-ULP-Bluetooth.htm - last checked: 29.11.2011.