Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Anwendungen 2: Related Work SoSe 2011

Heiner Perrey

Betreuer: Prof. Dr. Dirk Westhoff

Attacks on Wireless Networks

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Contents

1 Introduction

Bluetooth Low Energy (BLE) is fairly new technology and part of the new Bluetooth specification version 4.0 [5]. BLE is based on the classic Bluetooth standard (BT), but both are not directly compatible with one another. This means that new hardware is necessary for using BLE. Since BLE can run on a coin cell battery it is especially interesting for devices which have a highly limited power supply. As already depicted in [10] we are introducing a new protocol which combines BLE with Merkle's Puzzle (MP) to securely exchange a key (BLE+MP). Our work is based on the approach in [2]. We assume devices with asymmetric capabilities, which means that we are dealing with a fully functional device (FFD) and one or many reduced functional devices (RFD). An RFD may be a small sensor node which is not able to run complex key agreement protocols like Elliptic Curve Diffie-Hellman (ECDH). In our protocol we use MP to shift the workload nearly completely to the stronger device, thus decreasing the workload for the RFD. We propose a concept of adding MP as a supplementary security feature in the BLE specification. We point out some changes that have to be made to deploy the BLE+MP protocol. To keep these changes at a minimum we try to work as closely to the specification as possible. We propose to use the newly introduced advertisements in BLE to broadcast the potential keys according to MP.

Merkle's Puzzle is a procedure to exchange a key over an insecure channel [8]. It is especially interesting for devices with asymmetric capabilities, since the workload can be shifted nearly completely to the stronger device. Although MP only offers temporary security it is the basis for the Diffie-Hellman protocol [7]. The key exchange between the FFD and one or more RFD is shown in figure 1.1.[1] The FFD and the RFD want to agree upon a symmetric key. Therefore the FFD creates n puzzles of the form $P_i = E_{k_i}(P_{ID_i}, K_i)$, where K_i is one possible strong symmetric key to be exchanged, P_{ID_i} is the unique identifier for puzzle P_i, E is a encryption function and k_i a weak symmetric key the puzzle is encrypted with. Now the FFD broadcasts all puzzles over an insecure channel. The RFD only receives one puzzle by listening to the broadcast channel at a random time. The RFD now starts to break the weak key, k, of the puzzle using brute force. Note that the size of k must be chosen so that the RFD may quickly break the encryption. After the broadcast is done the RFD sends the P_{ID} retrieved from the puzzle back to the FFD. Since the FFD can map each identifier to the correct puzzle, it can determine the choice of the RFD. Now both share the same strong symmetric key K.

[1] The procedure is also depicted in [10].

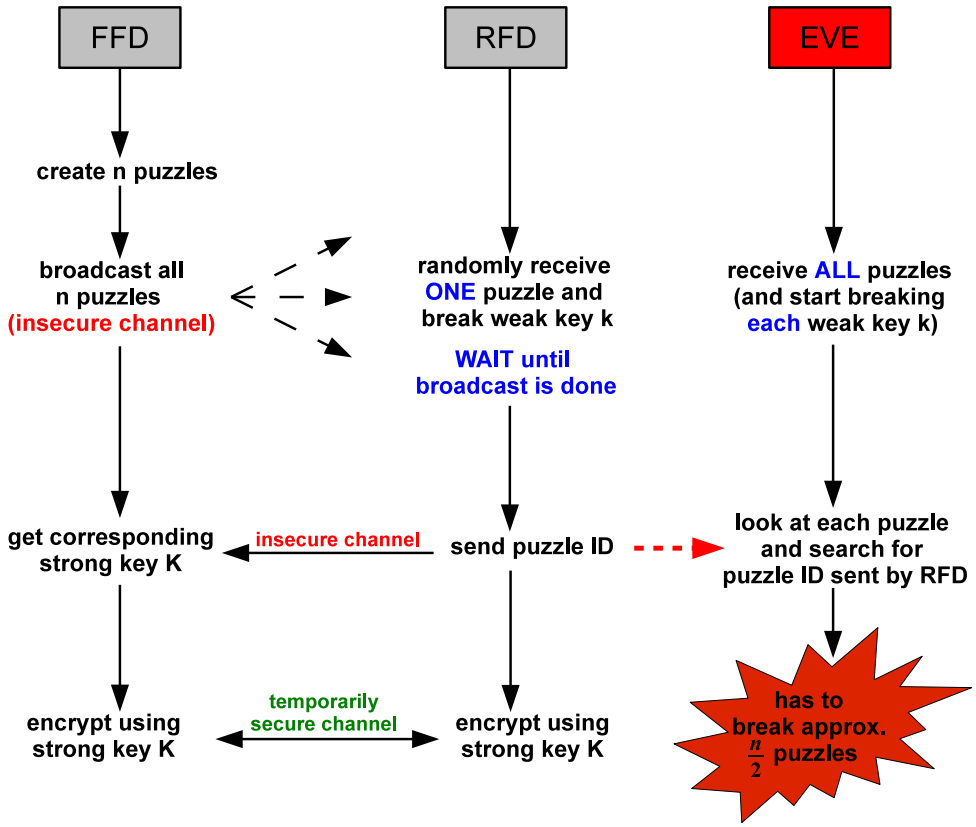

Figure 1.1: Outline of the Merkle's Puzzle protocol.

An eavesdropper Eve basically has the same knowledge as the RFD. She knows all puzzles and the P_{ID} that the RFD sent back to the FFD. Since the P_{ID} does not let Eve conclude which puzzle the RFD chose, she has to randomly break puzzles until she finds the right identifier. This implies that Merkle's Puzzle only offers temporary protection against an eavesdropper. Because she has to break an average of $\frac{n}{2}$ puzzles she has a quadratic complexity as opposed to the FFD and RFD which only have linear costs.[2]

The security of the MP protocol can be adjusted using the parameters n and the size of k. Therefore theoretically any level of security is achievable. In practice however the threshold value is defined by the capabilities of the FFD and RFD and by the time restrictions of the application.

[2] For performance analysis of BLE+MP see [10].

2 Related Work

This chapter gives an overview of the related work. First some studies concerning the key agreement for restricted devices are presented. Then a few works about BLE are introduced.

2.1 Security for Energy Restricted Devices

2.1.1 "Using Merkle's Puzzle for Key agreement with Low-end Devices" [2]

Frederik Armknecht and Dirk Westhoff depict a key agreement protocol for low end devices using Merkle's Puzzle [2]. As described in chapter 1 the work at hand follows up on a quite similar approach using BLE. The authors in [2] suggest a key agreement for sensor nodes (especially WBAN) using Zigbee. They consider a setup of devices with asymmetric capabilities (as depicted in chapter 1). By using Merkle's Puzzle the workload can be shifted to the stronger device.

Expecting BLE to become a trendsetting technology we propose to extend the idea of using Merkle's Puzzle with extremely restricted devices. Our work can be seen as a conceptual study. We want to point out necessary changes that have to be made in order to add MP to the BLE specification. Hereby we work as closely to the BLE document as possible.

2.1.2 Castelluccia et al. [3, 6]

"Shake Them Up!" [6]

Claude Castelluccia and Pars Mutaf present a technique to securely agree upon a key by shaking the devices [6]. For key agreement they use an optimized version of the *anonymous channel* introduced in [1]. In an anonymous channel the source cannot be identified. As depicted in figure 2.1, terminal A and B want to exchange a key. For the protocol to start terminal A sends a command message to B containing the size of the secret key k and the source address. In a reply message, B sends its address to A. In each round an empty message is sent in which the source and destination address is randomly set to the values A

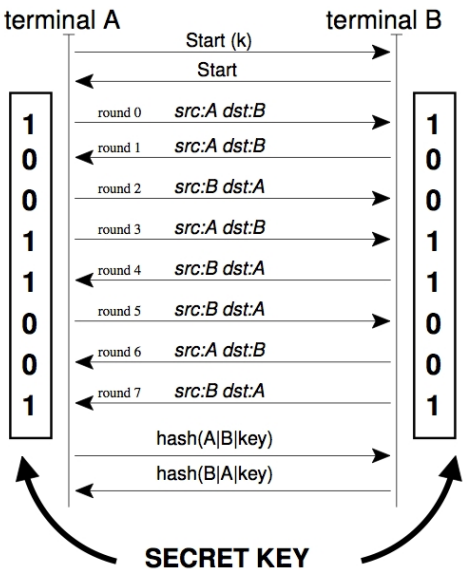

Figure 2.1: Key exchange with „Shake Them Up“ [6].

respectively B. The secret key consists of the boolean values whether the addresses have been set correctly. Since both know their own and the other party's address and they know whether or not they sent a specific message, they can distinguish between *true* and *false*. An adversary on the other hand can not tell whether the source and the destination have been set correctly and therefore does not know if the corresponding key bit is *true* (1) or *false* (0). After the agreement, both A and B compute the hash value of the secret key to ensure both sides share the same key bits. To prevent an attacker from using a signal power analysis to distinguish the source, the authors suggest to shake the devices during the key agreement. In their work they show that this procedure successfully keeps an adversary from determining the correct source address.

The authors acknowledge that user input is necessary in order to securely agree upon a key. They argue that shaking the devices during pairing is only a small effort and therefore a reasonable requirement. This being true for many applications, we do not want to consider any user input in our work. In our assumptions the sensor nodes may be spread out in an apartment or hospital for instance or the nodes may be installed permanently at places out of reach. Another issue they address is the use of an anonymous channel. This means that the real source address needs to be unknown to the attacker. Since Bluetooth uses time devision multiplexing access (TDMA), in which packets are sent on different channels for a period of time, the source could be exploited. Therefore they do not consider their protocol to be used with Bluetooth.

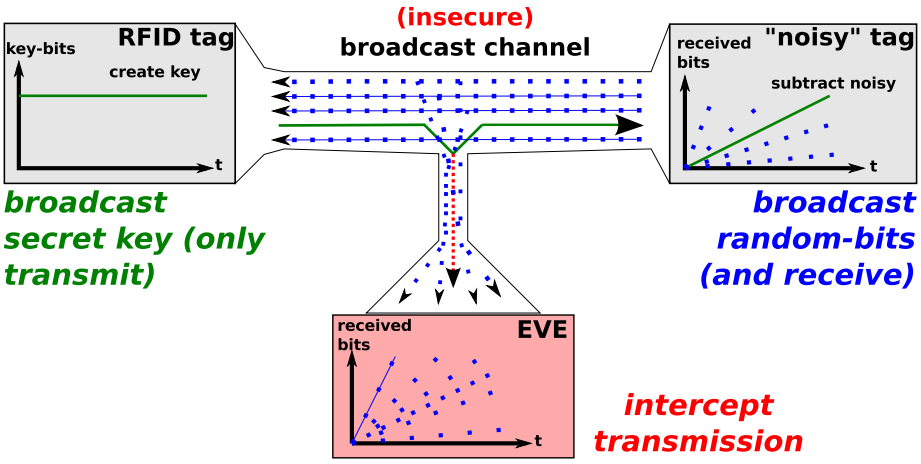

Figure 2.2: Key exchange between *RFID- & Noisy-tag*.

"Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags" [3]

Claude Castelluccia and Gildas Avoine present another approach similar to the one dipicted in section 2.1.2 [3]. They concentrate on the key agreement for RFID tags using a *noisy tag*. Figure 2.2 demonstrates the protocol. The RFID tag wants to share a secret key with the noisy tag (another RFID tag) over an insecure broadcast channel. To do so, the RFID tag creates a secret key. Both the RFID tag and the noisy tag need to synchronize before the key exchange. As soon as the RFID tag begins to send the key the noisy tag sends random bits synchronously. The noisy tag concurrently receives all data coming from the broadcast channel. The noisy tag will receive one data stream which consists of the key bits mixed with the random bits. To obtain the key the noisy tag simply has to subtract the generated noise from the data stream. Now both the RFID tag and the noisy tag share the same key.

A potential eavesdropper, Eve, will also receive all bits. Without the knowledge which bit has been randomly created and which belongs to the key, she cannot obtain the secret key. This protocol also comes with the precondition that the source stays anonymous. So as mentioned in section 2.1.2 this protocol also does not work with Bluetooth due to the use of TDMA.

2.1.3 "A review on the PKC-based security architecture for wireless sensor networks" [12]

Iftekhar Salam, Pardeep Kumar and HoonJae Lee suggest a more common approach for key exchange in wireless sensor networks (WSN). In [12] they optimize the procedure of exchang-

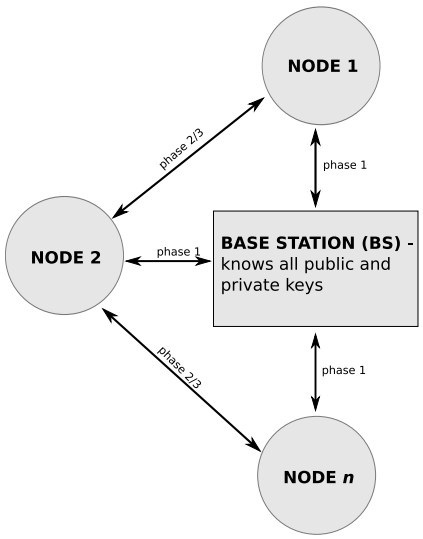

Figure 2.3: Key exchange using *pre-distribution scheme*.

ing a symmetric key via asymmetric key cryptography. They address the problem of bootstrapping each node of the WSN with a secret information for key exchange.

The pre-distribution scheme works as depicted in figure 2.3. The public key of the BS is precoded in each node. In phase 1 each node uses the base stations (BS) public key to contact the BS and receive their public/private key pair. This phase takes place before deployment of the WSN for instance in a hostile environment. After deployment in phase 2 the nodes only contact their direct neighbors to set up a bi-directional link. Once set up the nodes exchange their public key with each one-hop neighbor. In phase 3, a symmetric link key is exchanged using public/private key cryptography. Using this approach every node only needs to know the BS public key, its own public/private key and the public keys of its direct neighbors. Furthermore by using symmetric cryptography to encrypt the data transmissions more resources can be saved.

They consider limited devices functioning in a WSN. In our approach we argue from the point of view, that the sensor nodes are highly limited such that they are not able to run public/private cryptography. Furthermore by using MP we do not need a trusted third party.[1]

[1] The authors address some points in a review paper [13].

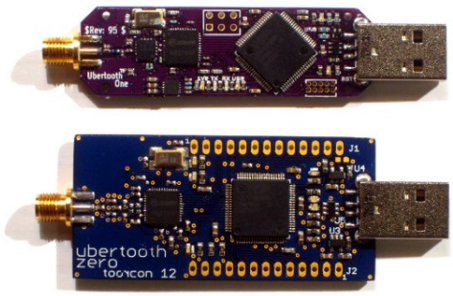

Figure 2.4: *Ubertooth Zero & One* [11].

2.1.4 Verbauwhede et al.

The authors in [15, 17] devote their work to the trade-off of computational and communication cost regarding the security and energy consumption of the devices. Their main focus lies on the electronic health sector. They argue that the security features e.g. for brain implants are vitally important, since they are directly linked to the patient's health. They also acknowledge that security always comes at the cost of more computational power and therefore energy consumption. Hence with a more powerful device the need for cooling increases. While considering the significance of security for a brain implant they also point out the lack of cooling possibilities for devices mounted inside or outside the human body. To address the issue of cooling and the computational restrictions of such devices the authors concentrate on energy efficient security functions. They compare different security protocols and discuss their energy consumption. Hereby their goal is to show some solutions and to present the problem of the communication and computation trade-off itself. In our work we argue similar to the authors in [15]. Hereby we are looking at the special case of having extremely limited devices where no asymmetric cryptography is applicable. We also discuss the power constraints of the sensors in use.

2.2 Bluetooth Low Energy

2.2.1 Bluetooth (LE) Monitoring Devices

In his blog Michael Ossmann demonstrates his development of Bluetooth monitoring devices [9]. His goal is to make Bluetooth monitoring devices affordable for basically anyone for less than 100 $ [14]. Therefore he developed the *Ubertooth Zero* and *Ubertooth One* (s. figure 2.4) to sensitize for an attack on Bluetooth.

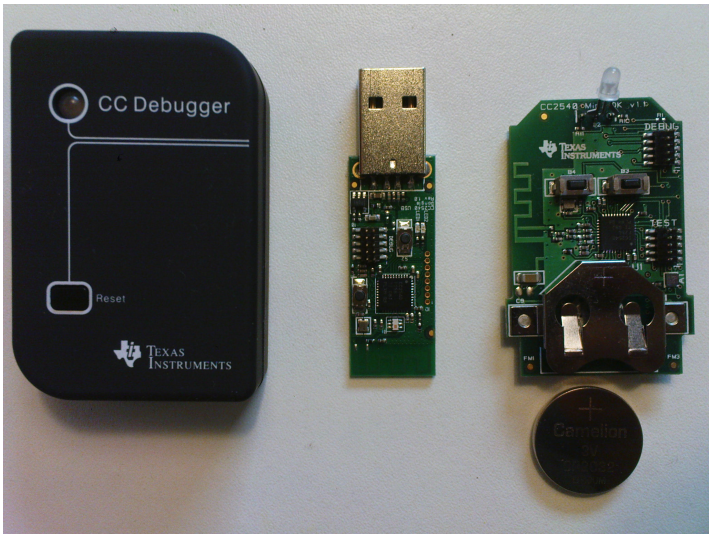

Figure 2.5: *Texas Instruments Bluetooth Low Energy CC2540 Mini Development Kit.*

He argues that for several reasons hardly any Bluetooth monitoring devices have been introduced. Therefore the security issue has been left unattended so far. To prevent this from happening to BLE he announced the development of a BLE monitoring device in [14].

Since we are assuming a passive attack on BLE, Ossmann's work can be seen as the basis for our attacker model. Through his work he shows that studies to prevent a passive (or active) attack are all the more important. In addition to that we may use his boards for analysis of our implementation.

2.2.2 "Sensornetzwerk mit Bluetooth Low Energy" [16]

In [16] Benjamin Tanner and Gabriel Gräni use BLE to set up a WSN with a different kind of sensors. They use the CC2540 Mini Development Kit by Texas Instruments (s. figure 2.5) for their implementation. The CC2540 Mini Development Kit offers a BLE USB-Dongle and keyfob as well as a BLE stack implementation, which can be downloaded from the Texas Instruments webpage [4]. In their thesis they give an introduction to BLE, the hardware (CC2540 and sensors), and the software they have written for their project. One of the areas they focus on is the energy restriction of sensor nodes. Therefore they present some ways and configurations to extend the battery life, e.g. by adapting the sending interval of the sensor measurements with respect to the power supply. They also use the newly introduced advertisements of BLE to broadcast the sensor measurements to the specific scanning device.

At the moment we are implementing the BLE+MP as a proof of concept. For hardware we use the CC2540 Mini Development Kit. Since in our concept we also intend to use the advertisements to broadcast the puzzles our software approach may be quite similar to the one presented in [16].

3 Overview and Outlook

In [2] the authors proposed to use Merkle's Puzzle for key agreement. They assume devices with asymmetric capabilities such that the FFD can take on the workload for the RFD. The RFD is extremely limited in its function and cannot run complex key agreement protocols like ECDH. Our work builds up on this idea by adding MP to the BLE specification and therefore is rather a conceptual work.

In [6] the authors securely agree upon a key by shaking the devices during the agreement. Hereby they can successfully prevent an attacker to use a power analysis. In [3] a noisy tag floods the broadcast channel with random bits while the key is transmitted by an RFID tag. By subtracting the noise from all received bits, the noisy tag can obtain the key. An attacker on the other hand can not differ between key-bits and noise. These being quite creative approaches they still require an anonymous channel to work properly. Therefore they are not suitable for the use with Bluetooth.

The authors in [12] propose a more common approach by using asymmetric cryptography to distribute a symmetric key in a WSN. Here a trusted third party is inevitable. They optimized the procedure by building a one-hop-network, so that each node only knows its direct neighbors. This being applicable for many WSN does not consider highly limited devices which are not able to run asymmetric key cryptography.

The authors in [15, 17] also show a major problem with security for restricted devices. They compare different approaches with respect to their energy consumption and therefore argue in quite a similar direction as we do.

Michael Ossmann [9] presents his development of Bluetooth monitoring devices. His open source project allows anybody to build such a device. Hereby he sensitizes for an exploit on Bluetooth. Showing that an passive attack on Bluetooth is fairly easy he depicts the importance for protocols that prevent such an attack. Also the authors in [16] experiment with BLE. Using the CC2540 Mini Development Kit they communicate with different types of sensors using advertisements. In our project we work with the same hardware.

Bibliography

[1] B. Alpern and F. Schneider. Key Exchange Using “Keyless Cryptography”. *inf. Process. Lett.*, pages 79–81, 1983.

[2] 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.

[3] G. Avoine and C. Castelluccia. Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags. *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference*, pages 289–299, April 2006.

[4] Bluetooth low energy software stack and tools. Texas Instruments. http://focus.ti.com/docs/toolsw/folders/print/ble-stack.html?DCMP=RF/IFANDZIGBEE& - last checked: 17.08.2011.

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

[6] C. Castelluccia and P. Mutaf. Shake them up! A movement-based pairing protocol for CPU-constrained devices. *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pages 51–64, December 2005.

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

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

[9] mossmann’s blog. webpage. http://ossmann.blogspot.com/ - last checked: 13.05.2011.

[10] H. Perrey, O. Ugus, and D. Westhoff. Security Enhancement for Bluetooth Low Energy with Merkle’s Puzzle. *Poster at Fourth ACM Conference on Wireless Network Security*, June 2011.

[11] Project Ubertooth. webpage. http://ubertooth.sourceforge.net/ - last checked: 20.05.2011.

[12] I. Salam, P. Kumar, and HoonJae Lee. An Efficient Key Pre-distribution Scheme for Wireless Sensor Network Using Public Key Cryptography. *Sixth International Conference on Networked Computing and Advanced Information Management (NCM), 2010*, pages 402 – 407, August 2010.

[13] I. Salam and HoonJae Lee. A review on the PKC-based security architecture for wireless sensor networks. In *2010 5th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 649 –652, December 2010.

[14] M. Ossmann: Building a Better Bluetooth Adapter. Shmoocon (talk). http://www.shmoocon.org/2011/videos/Ossmann-Bluetooth.m4v - last checked: 16.07.2011.

[15] D. Singelée, S. Seys, L. Batina, and I. Verbauwhede. The Communication and Computation Cost of Wireless Security - Extended Abstract. In *In Proceedings of the 4th ACM conference on Wireless network security (WiSec 2011)*, June 2011.

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

[17] Ingrid Verbauwhede. Low budget cryptography to enable wireless security. (Talk at) Fourth ACM Conference on Wireless Network Security, June 2011. http://www.sigsac.org/wisec/WiSec2011/WiSecKeynoteII.pdf - last checked: 15.07.2011.