Security Overview of Bluetooth

Dave Singelée, Bart Preneel

COSIC Internal Report

June, 2004

Abstract

In this paper, we give a short overview of the security architecture of Bluetooth. We will especially focus on the key exchange protocol in Bluetooth. This is the most important security critical part of the security architecture. Unfortunately, there are a lot of security flaws in the Bluetooth standard. Some are rather theoretical, but most of the problems can be exploited by an attacker. An extensive overview of the security flaws in Bluetooth will be given in this paper. Some of these security problems, e.g. the Bluesnarf attack, were only discovered very recently.

1 Introduction

In February 1998, the *Bluetooth Special Interest Group (SIG)* [3] was founded by the leaders in the telecommunications and network industries: Ericsson, IBM, Intel, Nokia and Toshiba. In the next 6 years, a lot of companies joined the SIG and now there are already more than 3000 members. The major task of this trade association was creating the Bluetooth specification which describes how mobile phones, computers, PDAs, headsets and other mobile devices can communicate wireless with each other. In June 2000, the Bluetooth standard was included in *IEEE 802.15* [10], the *Wireless Personal Area Network (WPAN) Working Group.*

The Bluetooth wireless technology realizes a low cost short-range wireless voice- and data-connection through radio propagation. The maximal range is about 10 m. The Bluetooth wireless technology uses the 2.4 GHz band, which is unlicensed, and can be used by many other types of devices such as cordless phones, microwave ovens, WIFI [16] and baby monitors. Any device designed for use in an unlicensed band should be designed for robustness in the presence of interference, and the Bluetooth wireless technology has many features that provide such robustness (*spread spectrum and frequency hopping*). The theoretical maximum bandwidth is 1 Mbit/s. The real bandwidth is lower because error correction (*Forward Error Correction*) is used in Bluetooth). One of the main differences between Bluetooth and some other wireless technologies is the ability to connect different sort of devices (e.g., a mobile phone with a PDA).

It is possible to configure the "visibility" of a Bluetooth device. When a device is in *non-discoverable mode*, it does not respond to inquiries of other devices. When the device is in *limited discoverable mode*, it is discoverable only for a limited period of time, during temporary conditions or for a specific event. And finally, when it is in *general discoverable mode*, it is discoverable (visible) continuously.

Bluetooth also has a security architecture. The most important part of this architecture, in particular the key agreement protocol, is discussed in section 2. Unfortunately, there are some security weaknesses in the Bluetooth standard. An extensive overview of these security problems is described in section 3.

2 Key Agreement Protocol in Bluetooth

An overview of the entire (security) architecture of Bluetooth [4] would take far too many pages. That is why we will focus on the key agreement protocol [14]. This is certainly the most crucial part of the security architecture. Suppose that two Bluetooth devices (called A and B) want to communicate securely. Initially these devices don't have a common shared secret. That is why they will perform a key agreement protocol. The result is a *link key* and an *encryption key.* The latter is used as the key of the stream cipher E_0. The process of generating a shared secret is called *pairing* (two Bluetooth devices are paired when they share a key which can be used to communicate securely).

2.1 Generation of the unit key

When a Bluetooth device is turned on for the first time, it calculates a so-called *unit key.* This is a key that is unique for every device and it is almost

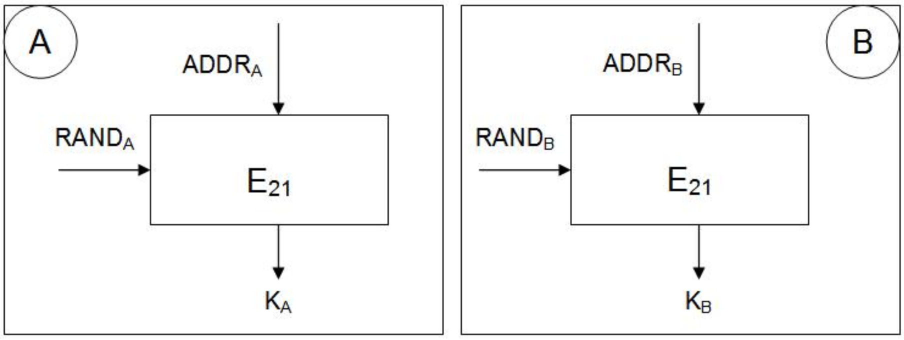

Figure 1: Generation of the unit key

never changed (a user can change the unit key, but this will not be done in practice). It is stored in non-volatile memory. The unit key is only used if one of the devices does not have enough memory to store session keys (see also Sect. 2.4 for more details). It is generated as follows: first, the device calculates a random number $RAND$. The unit key is based upon this random number and the Bluetooth address (which is a factory-established parameter unique for every device). This procedure is shown in Figure 1. More information about the key generation algorithm E_{21} can be found in the Bluetooth standard [4].

2.2 Generation of the initialization key

The Bluetooth devices still do not share a session key. This will be done in different steps. First, an *initialization key* is generated. This temporary key is a function of a random number IN_RAND (which is generated by the device that initiated the communication (let's suppose that A did this) and sent to the other device (B)), a shared PIN and the length L of the PIN. The PIN should be entered in both devices. The length of the PIN can be chosen between 8 and 128 bits. Typically, it consists of 4 decimal digits. If one of the devices does not have an input interface, a fixed PIN is used (often, the default value is 0000). This procedure is shown in Fig. 2. The result is a temporary shared key (the *initialization key*). More information about the key generation algorithm E_{22} can be found in the Bluetooth standard [4].

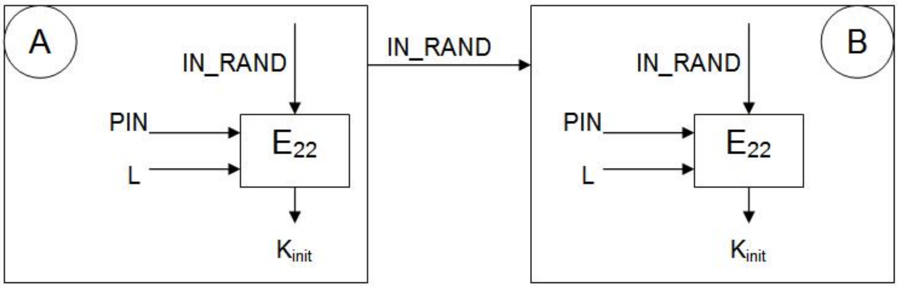

Figure 2: Generation of the initialization key

2.3 Mutual authentication

Each time a new shared key is generated (an initialization key (§2.2) or a link key (§2.4)), both devices perform a mutual authentication protocol. The authentication scheme is based upon a challenge-response protocol. This protocol is performed twice. First, B authenticates itself to A, as shown in Fig. 3. If this authentication is successful, the roles are switched (B becomes the verifier and A the prover). The authentication goes as follows. After B has sent its Bluetooth address, A generates a random number AU_RAND and sends this to B. This random number is called the *challenge*. Both devices now calculate a *response* $SRES = E_1(ADDR_B, K_{link}, AU\_RAND)$. K_{link} is the shared key, this could be an initialization key or a link key. The authentication function E_1 is a computationally secure authentication code. It uses the encryption function SAFER+. The algorithm is an enhanced version of an existing 64-bit block cipher SAFER-SK128, and it is freely available (more information can be found in the Bluetooth standard [4]). B sends its response to A. If this response corresponds to the value that A has calculated, then the authentication is successful. The authentication scheme also has the value ACO (*Authenticated Ciphering Offset*) as result. This value is used for the generation of the encryption key (see Sect. 2.5 for more details).

2.4 Generation of the link key

Both devices now share an initialization key. This key will be used to agree on a new, semi-permanent key (called the *link key*). The link key will be stored on both devices so that it can be used for future communication. Depending on the memory constraints of both devices, the link key can be the unit key

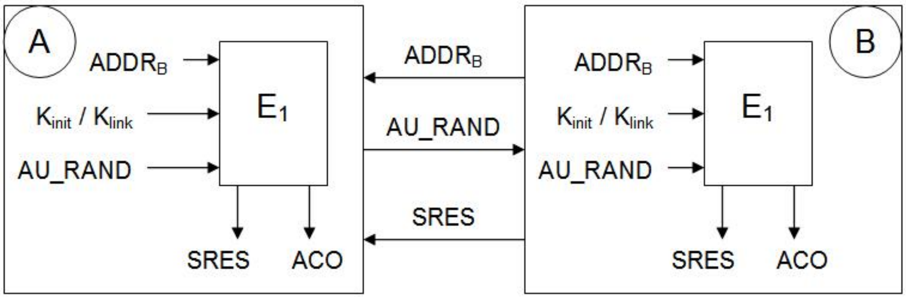

Figure 3: Mutual authentication protocol

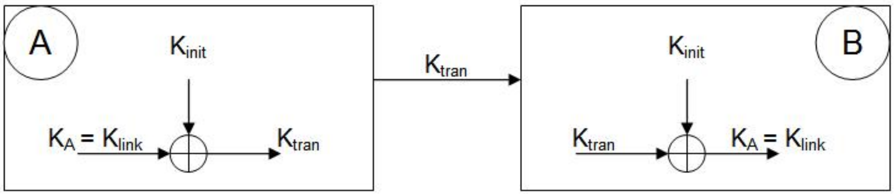

Figure 4: The link key is a unit key

of the memory-constrained device (Figure 4) or a combination key derived from the input of both devices (Figure 5).

If the unit key of device A is the link key, it is transmitted encrypted from A to B. This encryption is done by XOR'ing with the initialization key, as shown in figure 4.

If the link key is a combination key, then both devices first generate a random number LK_RAND. These random numbers are encrypted with the initialization key and sent to the other device. Now they both can calculate LK_K_A and LK_K_B. The combination key K_{AB} is the XOR of LK_K_A and LK_K_B. This is shown in Figure 5. The key generation algorithm E_{21} is the same as the algorithm used for the generation of the unit key. After the generation of the link key, the old temporary initialization key is definitively discarded and a mutual authentication is started with the exchanged link key (this is already discussed in section 2.3).

The procedure shown in Figure 5 is also executed when a new link key is calculated. The only difference is that the random numbers LK_RAND are encrypted with the old link key. The result is a new link key which will replace the old link key (this old link key will be discarded).

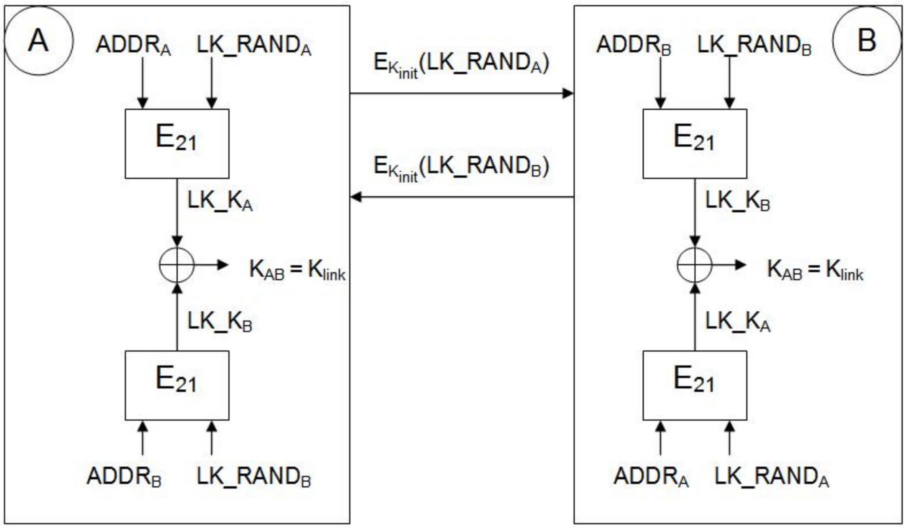

Figure 5: The link key is a combination key

2.5 Generation of the encryption key

After a successful generation of the link key and mutual authentication protocol, the encryption key can be generated. A generates a random number EN_RAND_A and sends this to B. Both devices generate the encryption key $K_C = E_3(EN\_RAND_A, K_{link}, COF)$. This is shown in Figure 6. More information about the key generation algorithm E_3 can be found in the Bluetooth standard [4]. The COF value (*Ciphering Offset Number*) is the ACO value which was generated during the mutual authentication protocol. There is however one exception. If the encryption key is used to broadcast encrypted messages, the COF is equal to the concatenation (denoted by $||$) of the Bluetooth address of the sender for broadcast encrypted communication (so $COF = (ADDR||ADDR)$). The encryption key K_C can reduced in length to an encryption key K'_C if necessary.

2.6 Generation of the key stream

Finally, the encryption key K_c (or the length-reduced key K'_C) is fed to the encryption scheme E_0 together with the Bluetooth address and the clock of the master. The result is the key stream K_{cipher}. The master clock is used in order to make the key stream harder to guess. This is shown in Figure 7. The key stream K_{cipher} is XOR'd with the data that has to be encrypted

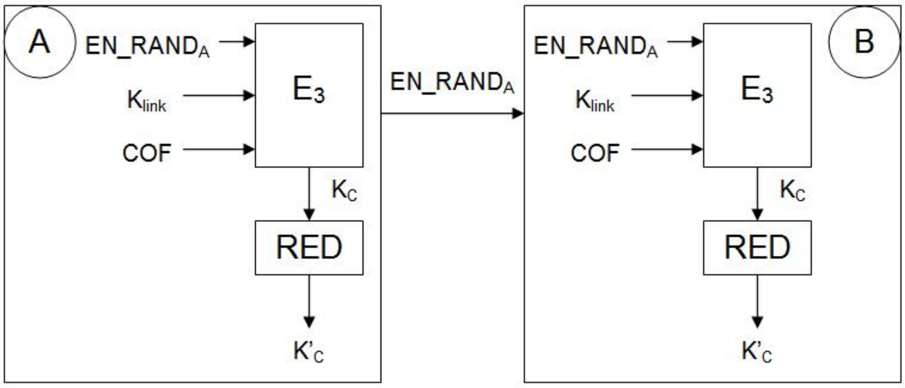

Figure 6: Generation of the encryption key

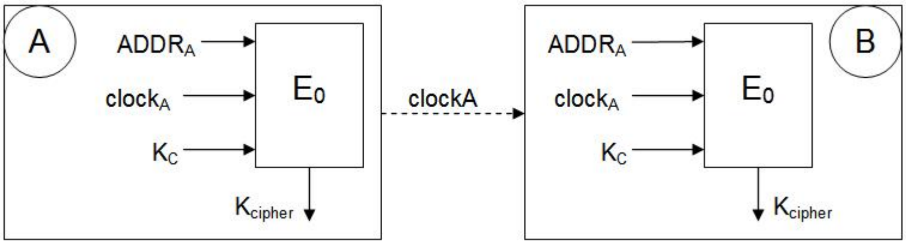

Figure 7: Generation of the key stream

(see Figure 8).

3 Security Weaknesses

There are many security weaknesses in the Bluetooth standard [13]. Some of these problems can very easily be exploited by an attacker, other security weaknesses are rather theoretical. An extensive overview of the most important problems will now be given.

3.1 Security depends on security of PIN

The initialization key is a function of a random number IN_RAND, a shared PIN and the length L of the PIN. The random number is sent in clear and hence known by an attacker that is present during the initialization phase. Note that it is not so difficult for an attacker to obtain this random number.

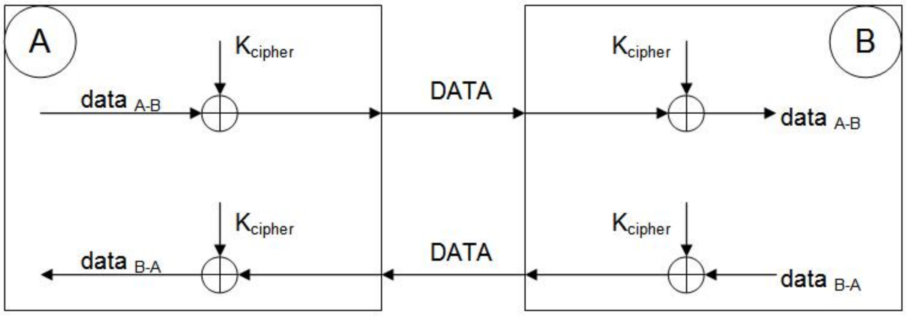

Figure 8: Encryption of the data

He can place some (small) devices near the two Bluetooth devices that are going to be paired or even place a small sensor on one of the devices. This means that only the PIN is a secret value, all the rest is public. If an attacker obtains the PIN, he knows the initialization key. It even gets worse! Since all the other keys are derived from the initialization key, they also will be known by the attacker. The security of the keys depends on the security of the PIN. If it is too short or weak (e.g., 0000), it is very easy for an attacker to guess the PIN.

Note that it is always possible to guess the PIN. The reason is that a mutual authentication protocol is executed after the generation of the initialization key. If an attacker observes this protocol, he obtains a challenge and the corresponding response. It is now very easy to perform a brute force attack. The attacker tries every PIN and calculates for every PIN the corresponding response. When the calculated response is equal to the observed response, the correct PIN is used. The shorter the PIN, the faster this brute force attack can be executed.

Sometimes a fixed PIN is used (the default value is 0000) or the PIN is sent in clear to the other device. In those cases, the PIN is publicly known and the keys only depend on public values. It is then trivial for an attacker to obtain the secret keys. This certainly has to be avoided in security-sensitive applications.

3.2 Unit key

The unit key is used if one of the Bluetooth devices does not have enough memory to store session keys. This key is stored in non-volatile memory and almost never changed. As already described in section 2.4, the unit key is sent encrypted (with the initialization key) to the other device. This is not very secure! Suppose A has sent its unit key to device B. The result is that B now knows the key of A and can use this key itself. B can send this unit key to C and impersonate itself as A. It is impossible for C to detect this impersonation attack. This is why the use of unit keys should be avoided.

3.3 Encryption algorithm

Bluetooth uses the encryption algorithm E_0. This stream cipher has some security flaws [6, 8, 9, 12]. There is however no efficient attack known on E_0. Several attacks on E_0 are published, but most of these attacks do not work on the algorithm which implements E_0 in Bluetooth.

The attacks with the lowest complexity are the algebraic attacks [7]. E_0 is vulnerable to algebraic attacks because of the possibility to recover the initial value by solving a system of non-linear equations of degree 4 over the finite field $GF(2)$. This system can be transformed by linearization into a system of linear independent equations with at most 2^{23} unknowns. Fortunately, this attack does not work in Bluetooth because it needs a long key stream during the initialization and E_0 in Bluetooth only uses small packets (the payload ranges from zero to a maximum of 2745 bits [4]).

There is however an attack which can be implemented on the E_0 algorithm in Bluetooth. Golic [11] has found an attack on the Bluetooth stream cipher that can reconstruct the 128-bit secret key with complexity about 2^{70} from about 45 initializations. In the precomputation stage, a database of about 2^{80} 103-bit words has to be sorted out. The attack uses a general linear iterative cryptanalysis method for solving binary systems of approximate linear equations.

3.4 Denial of service attacks

Mobile networks are vulnerable to denial of service attacks [1]. They consist of mobile devices and these devices are often battery fed. Bluetooth is no exception. An attacker can send dummy messages to a mobile device. When this device receives a message (a real of a fake one), it consumes some computation (and battery) power. After some time, all battery power will be consumed and the device won't be available anymore. This exhaustion of the battery power is called the *sleep deprivation attack*. There are a lot more denial of service attacks. The attacker can try to interfere with the radio propagation. Bluetooth uses the 2.4 GHz ISM band, which is also used by some other mobile networks (e.g., WIFI [16]). To avoid the interference caused by other mobile networks or an attacker, *frequency hopping* and *spread spectrum* are used in Bluetooth.

There are also some denial of service attacks caused by implementation decisions. An example is the *black list* which is used during the mutual authentication procedure. To avoid that a device would start a mutual authentication procedure over and over again, each device has a black list with the Bluetooth addresses of the devices which failed to authenticate themselves correctly. These devices can not start an authentication procedure during some period. This period will be increased exponentially (until a certain upper limit is reached) if the authentication fails again. The black list is used to avoid a denial of service attack (successive wrong authentication procedures), but in fact opens the door for other DoS attacks [5]. An attacker can try to authenticate to device A, but change every time its address. All these authentication attempts will fail and the black list of A will become quite large. If there is no upper limit on this black list, the entire memory of A will be filled with the entries of the black list and device A will crash. This is not the only DoS attack. Suppose device B wants to authenticate to A. After A has sent a random number (*the challenge*) to B (this is the first step in the authentication procedure), the attacker sends a wrong *response* to A using the Bluetooth address of B. The authentication will fail, B will be put on the black list of A and hence the (correct) response of B will be ignored by A. The attacker keeps repeating this attack and B will never be able to authenticate successful to A.

3.5 Location Privacy

When two or more Bluetooth devices are communicating, the transmitted packets always contain the Bluetooth address of the sender and the receiver. When an attacker eavesdrops on the transmitted data, he knows the Bluetooth addresses of the devices which were communicating (the attacker can do this by placing a small device near the two Bluetooth devices). This way, the attacker can keep track of the place and the time these two devices were communicating. It is also quite probable that the two devices are from the same user (most of the communication takes place between devices of the same owner).

This is a violation of the privacy of the user. The location information can be sold to other persons and used for location dependent commercial advertisements (e.g., a shop can send advertisements to all the users which are near the shop). It should be possible for the user to decide himself when his location is revealed and when not.

3.6 Bluesnarf attack

Recently, the *Bluesnarf attack* [15] was discovered. It is possible, on some mobile phones, to connect to the device without alerting the owner of the target device of the request, and gain access to restricted portions of the stored data in the phone, including the entire phonebook (and any image or other data associated with the entries), calendar, realtime clock, business card, properties, change log, IMEI (*International Mobile Equipment Identity*, which uniquely identifies the phone to the mobile network, and is used in illegal "phone cloning"), ... This is normally only possible if the device is in *discoverable mode*, but there are tools available on the Internet that allow even this safety net to be bypassed.

The Bluesnarf attack can also be extended by combining it with a *backdoor attack* [15]. The result of this combined attack is that not only the private data of the mobile phone can data be retrieved, but other services, such as access to the Internet, WAP [17] and GPRS [17] gateways or even sending an SMS are available for the attacker without the owner's knowledge.

3.7 Bluejacking

When two Bluetooth devices are paired, these devices will send their "name" to each other. This user defined name can be up to 248 characters and will be displayed on the output-interface of the other device. The *Bluejacking attack* [2] exploits this name to send advertisements to Bluetooth devices. The name of the sender is the advertisement itself (e.g., buy product X now). This is not really a security problem, but it can become a big problem (e.g., think of the amount of SPAM emails a user receives daily).

3.8 Other security problems

There are also some security problems in the challenge-response protocol. Another security flaw is that there is no integrity check on the Bluetooth packets. An attacker can always modify or replace a transmitted Bluetooth packet. Man-in-the-middle attacks are also not prevented in Bluetooth. And to make things even worse, a user can switch off security. Often, the default configuration is no security at all. This certainly has to be avoided.

4 Conclusion and Future Work

We have given a short overview of the key exchange protocol in Bluetooth. This is the most important security critical part of the security architecture. Unfortunately, there are many security flaws in the Bluetooth standard. Some are rather theoretical (e.g., the problems with the encryption algorithm E_0), but most of the problems can be exploited by an attacker. An extensive overview of the security flaws in Bluetooth is given in this paper. Recently, the new Bluetooth standard was published, but we regret that all security problems are still present in this new standard.

When Bluetooth is used in security critical applications, the users should be aware that extra cryptographic methods have to be used to be secure. Research is necessary on a new security architecture that solves the security problems in Bluetooth. This is not an easy problem to solve, because Bluetooth is often used by memory- and energy-constrained mobile devices.

Acknowledgements

Dave Singelée is funded by a research grant of the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT). This work was also supported by the Concerted Research Action (GOA) Mefisto-2000/06 of the Flemish Government.

References

[1] J. Bellardo and S. Savage. 802.11 Denial–of–Service Attacks: Real Vulnerabilities and Practical Solutions. In *Proceedings of the 12th USENIX Security Symposium*, pages 15–28, 2003.

[2] Bluejacking. `http://www.bluejackq.com/`.

[3] Bluetooth Special Interest Group. `http://www.bluetooth.com/`.

[4] Bluetooth Specification. `https://www.bluetooth.org/spec/`.

[5] C. Candolin. Security issues for wearable computing and bluetooth technology. `http://www.tml.hut.fi/~candolin/Publications/BT/`, 2000.

[6] C. De Cannière, T. Johansson and B. Preneel. Cryptoanalysis of the bluetooth stream cipher. COSIC Internal Report, 2000.

[7] N. Courtois and W. Meier. Algebraic Attacks on Stream Ciphers with Linear Feedback. In *Advances in Cryptology - EUROCRYPT 2003*, Lecture Notes in Computer Science, LNCS 2656, pages 345–359. Springer-Verlag, 2003.

[8] S. Fluhrer and S. Lucks. Analysis of the E0 Encryption System. In *8th Annual International Workshop of Selected Areas in Cryptography (SAC) 2001*, Lecture Notes in Computer Science, LNCS 2259, pages 38–48. Springer-Verlag, 2001.

[9] M. Hermelin and K. Nyberg. Correlation Properties of the Bluetooth Combiner Generator. In *Proceedings of the 2nd International Conference on Information Security and Cryptology (ICISC '99)*, Lecture Notes in Computer Science, LNCS 1787, pages 17–29. Springer-Verlag, 1999.

[10] IEEE 802.15, the Wireless Personal Area Network Working Group. `http://www.ieee802.org/15/`.

[11] V. Bagini J. Golic and G. Morgari. Linear Cryptanalysis of Bluetooth Stream Cipher. In *Advances in Cryptology - EUROCRYPT 2002*, Lecture Notes in Computer Science, LNCS 2332, pages 238–255. Springer-Verlag, 2002.

[12] J. Lano and B. Preneel. Some ideas on the resynchronisation attack. COSIC Internal Report, 2004.

[13] M. Jakobsson and S. Wetzel. Security Weaknesses in Bluetooth. In *Proceedings of the Cryptographer's Track at the RSA Conference (CT–RSA '01)*, Lecture Notes in Computer Science, LNCS 2020, pages 176–191. Springer-Verlag, 2001.

[14] G. Lamm, G. Falauto, J. Estrada, and J. Gadiyaram. Security Attacks against Bluetooth Wireless Networks. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, pages 265–272. U.S. Military Academy, West Point, NY, June 2001.

[15] A. Laurie and B. Laurie. Serious flaws in bluetooth security lead to disclosure of personal data. `http://bluestumbler.org`.

[16] The WI–FI alliance. `http://www.wi-fi.org/`.

[17] WAP Forum. `http://www.wapforum.org/`.