Proceedings of the International Conference on Information Technology and Multimedia at UNITEN (ICIMU 2001)
Recent Advances and Future Trends in Information Technology and Multimedia
13th – 15th August 2001, Malaysia

A Proposed Bluetooth Service-level Security

Fathi Taibi Mazliza Othman

Faculty of Computer Science and Information Technology
University Malaya, 50603 Kuala Lumpur

fathi@siswazah.fsktm.um.edu.my
mazliza@fsktm.um.edu.my

Abstract

Bluetooth security has become increasingly important since Bluetooth is going to be used as a standard technology in wireless personal communication. This is especially true for Bluetooth applications that require strong security policies, e.g. mobile commerce applications. In this paper, a Bluetooth service-level security that allows easy implementation of flexible access policies is proposed. Security is enforced based on service security level and device trust level.

Keywords: Bluetooth, service-level security.

1. Introduction

Formed in February 1998 by mobile telephony and computing leaders Ericsson, IBM, Intel, Nokia, and Toshiba, the Bluetooth Special Interest Group (SIG) has designed a technology specification allowing the development of interactive services and applications over interoperable radio modules and data communication protocols. Bluetooth can revolutionise wireless connectivity for personal and business mobile devices, enabling voice and data communication via short-range radio links, and allowing users to connect wirelessly to a wide range of devices easily and quickly. Bluetooth can be used in several scenarios like file transfer, access to Internet or data synchronisation between devices.

The use of Bluetooth as a standard in wireless personal communication depends on its security robustness. Bluetooth security is maintained using authentication of devices (verification of who is at the other end of the link) and data encryption. Bluetooth authentication is a challenge-response scheme based on a shared secret link key. To encrypt data, Bluetooth uses a symmetric encryption key derived from the link key and a stream cipher algorithm. In the Bluetooth specification (profiles part), three security modes are defined:

Security mode 1 (a non-secure mode): When a Bluetooth device is in security mode 1, it shall never initiate any security procedure, i.e. it shall never send LMP-au-rand (the link manager protocol data unit containing a 128-bit random number as a challenge in the authentication procedure), LMP-in-rand (the link manager protocol data unit containing a 128-bit random number used in the initialisation process) or LMP-encryption-mode-req (the link manager protocol data unit containing a request of the encryption mode that can be used).

Security mode 2 (a service-level enforced security mode): When a Bluetooth device is in security mode 2, it shall not initiate any security procedure before a channel establishment request (L2CAP-connetreq) has been received or a channel establishment procedure has been initiated by itself.

Security mode 3 (a link level enforced security mode): When a Bluetooth device is in security mode 3 it shall initiate security procedures before it sends LMP-link-setup-complete [1].

In this paper, Bluetooth link level security is presented and a Bluetooth service-level security architecture is proposed. Bluetooth link level security is not flexible, i.e. the same security procedures are applied to all devices without any consideration of the services requested. Flexibility, efficiency and robustness are the bases of the proposed

security; security procedures are enforced regarding to services and devices security levels.

In section 2, we present Bluetooth link level security and the procedures of generating the different keys used to maintain security. In section 3, we propose a Bluetooth service level security architecture. Section 4 explains the actions taken by the security manager when receiving an access request, and section 5 discusses the access check algorithms. Finally, section 6 concludes.

2. Bluetooth link level security

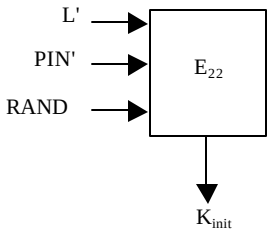
Four entities are used to maintain security in mode 3:

BD-ADDR: A unique 48-bit address field for each Bluetooth unit. BD-ADDR (Bluetooh Device Address) is publicly known (can be obtained via an inquiry routine by a Bluetooth unit).

RAND: Each Bluetooth unit has a random number generator. Random numbers (128 bits in length) are used during the authentication, and in the generation of the link and encryption keys. They must be non-repeating and randomly generated (must not be possible to predict their values).

Link key: A 128-bit private key shared between two or more parties. It is used for all security transactions between them. The link key may be a temporary key (an initialization key or a master key) or a semi permanent key (a unit key or a combination key).

Encryption key: A private key whose length is between 8 and 128 bits. A new encryption key is generated each time encryption is activated.



- PIN' is the PIN code augmented with the BD-ADDR of the claimant unit.
- L' is the length of PIN' (in byte)
- RAND is a random number generated by the verifier unit
- K_{init} is a 128 bit key

FIGURE 1: GENERATION OF THE INITIALIZATION KEY

2.1 Generation of the initialization key

The initialization key is used during the initialization process when no link key has been defined and exchanged. The initialization key protects the transfer of the initialization parameters. This key is derived by the E_{22} algorithm from a random number (generated by the verifier unit), L-octet PIN (Personal Identification Number) code augmented with the BD-ADDR of the claimant unit. The communication is initiated by the claimant (initiator) unit who requests connection to a verifier (acceptor) unit.

The PIN can be a fixed number provided by the Bluetooth unit (e.g. where there is no Man-machine Interface (MMI)) [4]. Alternatively, the PIN can be selected arbitrarily by the user, and entered in both units and the values must match. The latter procedure is used when both units have an MMI, e.g. a phone and a laptop. Entering a PIN in both units is more secure than using a fixed PIN. For many applications, the PIN code is a relatively short string of numbers. Typically it may consist of four decimal digits. For more sensitive situations, the PIN code may be chosen to be of any length from 1 to 16 octets. In this case, the units have to exchange PIN codes not through mechanical (human) interaction, but rather through means supported by software at the application layer, e.g. this can be a Diffie-Hellman key agreement.

2.2 Generation of the unit key

A unit key is generated when a Bluetooth unit is in operation for the first time. It is generated by the E_{21} algorithm with the BD-ADDR of the unit and a random number (RAND) as input. Once created, the unit key is stored in non-volatile memory and (almost) never changes. A unit key is 128 bits in length. Figure 2 shows the generation of the unit key of a unit A.

In cases where one of the units has restricted memory capabilities, its unit key can be used as a link key (since this unit only has to remember its own unit key) [2]. In figure 3, the unit key of unit A, K_A, is being used as a link key for the connection A-B. Unit A sends its unit key K_A protected by XOR-ing it with the initialization key K_{init}.

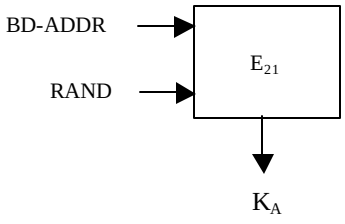


FIGURE 2: GENERATION OF THE UNIT KEY

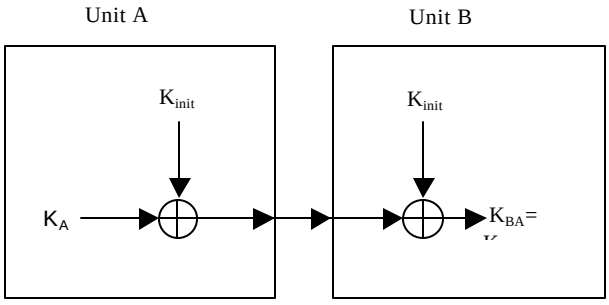


FIGURE 3: UNIT KEY EXCHANGE

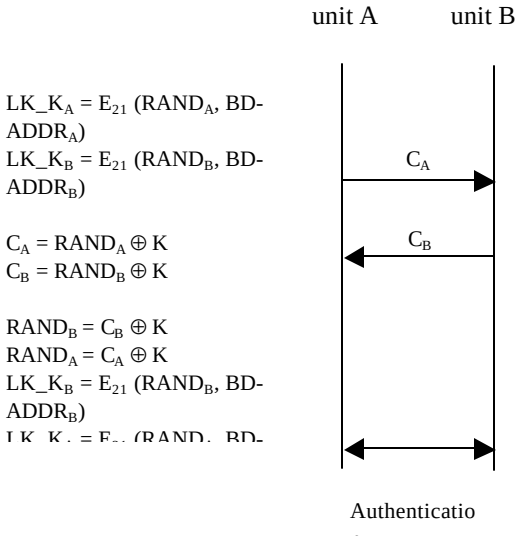


FIGURE 4: GENERATING A COMBINATION KEY

2.3 Generation of the combination key

The combination key is a combination of two numbers generated in unit A and B, respectively. First, each unit generates a random number (e.g. $RAND_A$ and $RAND_B$). Then using the E_{21} algorithm with the random number generated, and its own BD-ADDR, two random numbers $LK\_K_A = E_{21}$ ($RAND_A$, BD-$ADDR_A$) and $LK\_K_B = E_{21}$ ($RAND_B$, BD-$ADDR_B$) are created in unit A and unit B, respectively. Then the two numbers $RAND_A$ and $RAND_B$ are exchanged securely (protected by XOR-ing with the current link key K). After that unit A calculates LK-K_B and unit B calculates LK-K_A. Finally the combination key is calculated by each unit ($K_{AB} = K_{BA} = LK\text{-}K_A \oplus LK\text{-}K_B$).

2.4 Generation of the master key

The master key K_{master} is a link key used only during the current session. It replaces the original link key only temporarily and is used when the master wants to reach more than two Bluetooth units simultaneously using the same key [2].

First, the master creates a new link key from two 128-bit random numbers, RAND1 and RAND2, using the E_{22} algorithm: $K_{master} = E_{22}$ (RAND1, RAND2, 16). Then a third random number, (RAND), is generated and transmitted to the slave. After that, using the E_{22} algorithm with the current link key (K) and RAND as input, both the master and the slave computes a 128-bit overlay: OVL = E_{22} (K, RAND, 16). Finally, the master sends a bit-wise XOR of K_{master} and the OVL. The slave, who knows OVL, recalculates K_{master}.

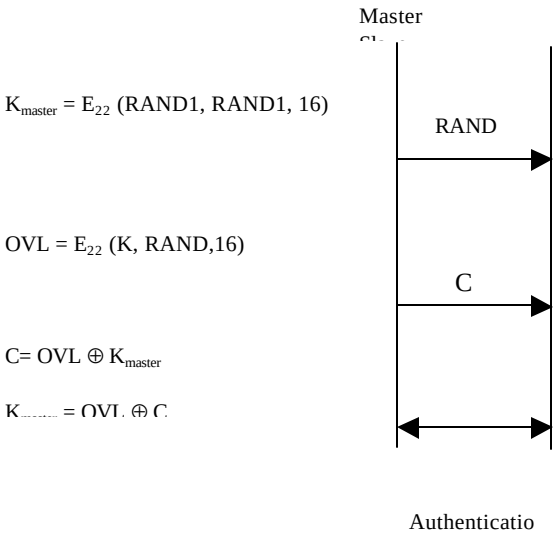


FIGURE 5: GENERATING A MASTER KEY

2.5 Authentication

Authentication is the process of verifying who is at the other end of the link [3]. Bluetooth authentication is a challenge-response scheme in which a claimant's knowledge of a secret key is checked using a symmetric secret key (K). First, the verifier challenges the claimant to authenticate a random number AU-RAND. Then both calculates a 32-bit SRES (Slave Response) and a 96-bit Authenticated Ciphering Offset (ACO) using E_1 (with BD-ADDR of the claimant, AU-RAND and the shared link key K as input). Then the claimant sends its calculated SRES' to the verifier who compares it to its calculated SRES. The authentication succeeds if SRES=SRES'.

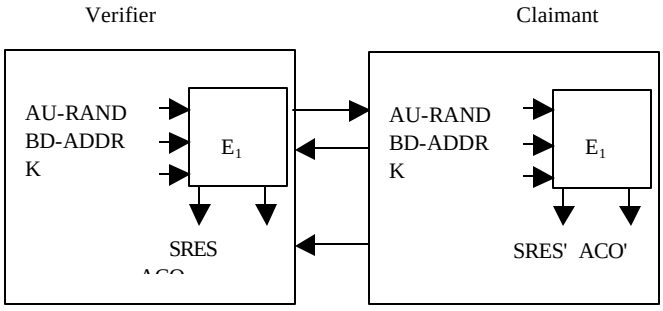


FIGURE 6: AUTHENTICATION PROCEDURE

2.6 Encryption

The encryption key is generated using the E_3 algorithm: first, the verifier generates a random number EN-RAND and sends it to the claimant. Then both calculate the encryption key Kc (8-128 bits) using the random number generated, the link key and a 96-bit Ciphering Offset (COF) which is either the ACO or the concatenation of the BD-ADDR of the claimant.

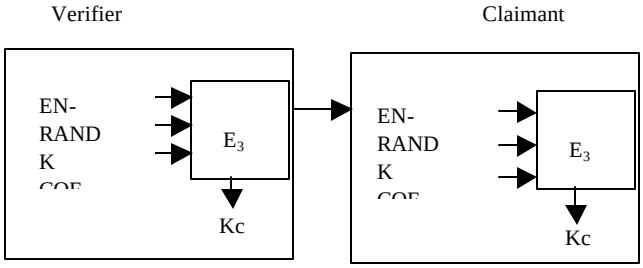


FIGURE 7: ENCRYPTION KEY GENERATION

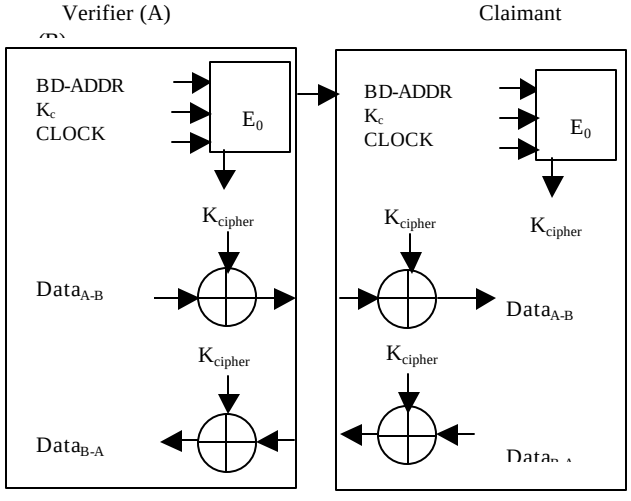


FIGURE 8: ENCRYPTED COMMUNICATION

The Bluetooth encryption system encrypts the payloads of the packets using a stream cipher E_0 that is synchronized for every payload.

The stream cipher E_0 has as input BD-ADDR of the verifier, the clock of the verifier (27 bits) and the encryption key Kc.

The shortcoming of the existing security feature are flexibility and efficiency. The same security procedures are applied without any consideration to the remote device trust level or the security level of the services requested. We address this shortcoming by proposing a Bluetooth service level security architecture where flexibility, efficiency and robustness are taken in consideration. The next section explains our proposed service-level security.

3. A proposed Bluetooth service-level security

The service-level security we are proposing is important because it addresses the shortcoming of Bluetooth link level security mode which leads to a significant refinement in the way that services are accessed (connection establishment) by remote Bluetooth devices.

Our proposed Bluetooth service-level security is based on a security manager that is responsible for granting or denying access to services. Granting of access depends on service security level and device trust level [5].

The security manager has to maintain security-related information about local services in a database known as a service database (SDB); it also has to maintain security-related information about remote devices in another database, known as a device database (DDB).

In order to perform its task, the security manager interacts with the Logical Link Control and Adaptation Protocol) L2CAP, the Host Controller Interface (HCI), the serial ports emulation protocol (RFCOMM) and the User Interface Entity (UIE) or the application acting on behalf of the user interface. The interfaces of the security manager with the other entities involved (L2CAP, HCI, RFCOMM, UIE) is outside the scope of this paper.
The following figure shows the proposed service level security architecture.

3.1 Services security level

Services security level is the need for authorization and/or authentication and/or encryption to access services from remote devices. Authorization is the process of deciding if a device x is allowed to access a service y. Authentication is the process of verifying who is at the other end of the link, and is achieved by the authentication procedure based on the stored link key or by pairing (entering a PIN) [6], [7].

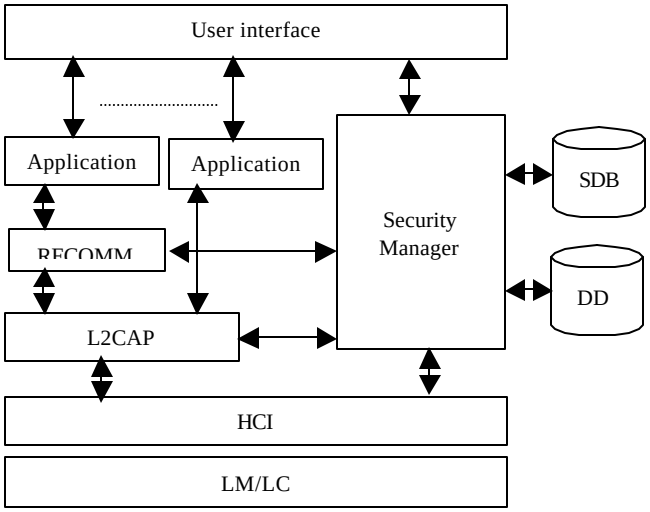


FIGURE 9: BLUETOOTH SEVICE LEVEL SECURITY ARCHITECTURE

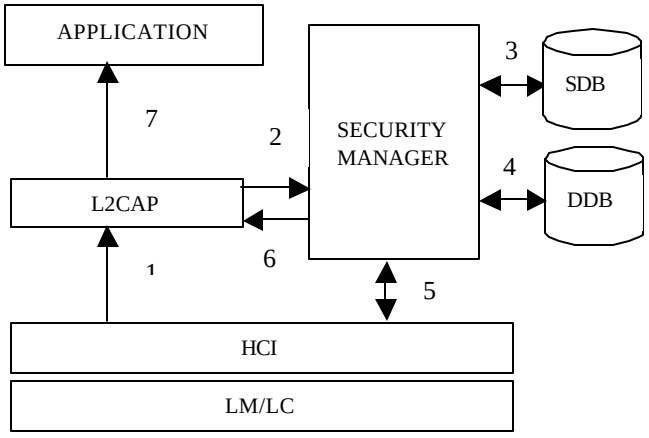


FIGURE 10: ACCESS REQUEST INFORMATION FLOW

3.2 Devices trust level

Devices are classified into 3 categories [5]:

Trusted devices: Devices that have been previously authenticated. A link key is stored and they are marked as "trusted" in devices database (DDB).

Untrusted devices: Devices have been previously authenticated, a link key is stored but they are not marked as trusted in DDB.

Unknown devices: No security information is available for these devices, they are considered as untrusted devices.

3.3 Services database

The security manager maintains a database with all the available services and their security levels. This security information is collected by querying the UIE or a special application acting on behalf of UIE. The information needed by the security manager is shown in Table1.

3.4 Devices database

The security manager has to maintain a database of all known devices and their trust levels. The number-of-failure attribute is used in the authorization procedure; it is compared to a limit number of failures allowed to grant authorization. Table 2 groups the attributes of this database.

TABLE 1: SERVICES DATABASE ATTRIBUTES

Attribute	Description
Service name	Service readable name (string).
Protocol/Service Multiplexer (PSM) value	Protocol service multiplexer value (2 bytes) .
Authorization-Requested	If this attribute is1 the authorization is required to access the service, 0 means the authorization is not requested.
Authentication-Requested	If this attribute is 1, authentication is required to access the service, otherwise it is not requested.
Encryption-Required	If this attribute is 1 the encrypted traffic must be used, if it is 0 the encryption is not required.

TABLE 2: DEVICES DATABASE ATTRIBUTES

Attribute	Description
BD-ADDR	48-bit IEEE address.
Device name	Device readable name (string up to 248 characters).
Trust level	Trust level=1 means the device is trusted, otherwise it is untrusted.
Link key	128-bit link key.
Number-of-failure	Number of authentication (or pairing) failures.

4. Access request information flow

Figure 10 illustrates the access request information flow, when L2CAP receives an access request to a specific application (1), it requests access from the security manager (2). First, the security manager looks up the services database (3) to determine the application security level; it also looks up the device database to determine the trust level of the device requesting access. Depending on the

security information obtained from the databases, the security manager starts an authorisation procedure or requests authentication (and/or encryption) from the host controller interface (HCI) (5). After that, the security manager may grant or deny access (6). If access is granted, L2CAP continues to set up the connection (7). The algorithms are discussed briefly in the next section.

5. Access check procedure

After determining the security procedure that must be applied to grant access, the security manager starts the access check procedure, a function ACCESS code is followed where ACCESS=TRUE means that access to the service is granted, ACCESS=FALSE means that access is denied:

ACCESS

Begin

ACCESS**:=True**;

If (authorisation **OR** authentication is required) **then**

Begin

AUTHENTICATION;

If (authentication failed) **then** ACCESS**:=false**;

If (authorisation is required) **then ATHORISATION;**

If (authorisation=false) **then** ACCESS**:=false**;

end;

If (encryption is required) **then ENCRYPTION;**

If (encryption failed) **then** ACCESS**:=false**;

End.

The authorisation procedure is only applied to untrusted devices, it is based on the number of authentication (or pairing) failures of a specific device, if this number is greater than the limit of failures that is allowed by the user then authorisation fails. The limit on the number of failures can be modified by UIE, also the UIE can assign a zero value to the number of failures allowed when setting up a new relationship.

AUTHORISATION

Begin

AUTHORISATION:=True;

If (device is not trusted) **then**

Begin

If (number-of-failure > limit) **then** AUTHORISATION:=false;

If (number-of-failure = 0) **then**

"The device is marked as trusted";

end;

End.

6 Conclusion

In this paper, a Bluetooth service-level security is proposed. The security proposed is based on a security manager that is responsible for granting access. It uses services of the existing link level security (authentication and encryption) in order to enforce security. The benefit of the security proposed is flexibility in terms of the security procedures applied compared to the link level security. Access to services that need an enforced security passes a rigorous access check procedure; other services can be accessed directly if no security measures (authorisation, authentication, and encryption) are needed. The security proposed is also efficient: authentication and encryption are enforced only when needed. In Bluetooth link level security, devices that fail the authentication (or pairing) procedure are not allowed to access any services even if the service requested does not need any security procedure - this is why we believe that using the Bluetooth service level security mode can really improve the robustness, the efficiency and the flexibility of Bluetooth security.

Further work on Bluetooth service-level security will be subject of our future papers.

References

[1] *Bluetooth specification, profile part, Bluetooth SIG.December 1999*. Internet: http://www.bluetooth.com/developer/specification

[2] *Bluetooth specification, core part, Bluetooth SIG. December 1999*. Internet: http://www.bluetooth.com/developer/specification

[3] Refik Molva, Didier Samfat, Gene Tsudik. December 1993. Authentication of Mobile Users. IEEE Network.

[4] *General information on Bluetooth*. 2001. Internet: http://www.mobileinfo.com/bluetooth

[5] Thomas Muller. July 1999. Bluetooth Security Architecture. Bluetooth white paper, **version 1**.

[6] Jaap Haartsen. February 2000. The Bluetooth Radio, System. IEEE Personal Communications.

[7] Riku Mettala. August 1999. Bluetooth Protocol Architecture. Bluetooth white paper, **version 1**.