A Bypassing Security Model for Anonymous Bluetooth Peers

Hahnsang Kim
INRIA, Sophia Antipolis, France
Email: Hahnsang.Kim@inria.fr

Walid Dabbous
INRIA, Sophia Antipolis, France
Email: Walid.Dabbous@inria.fr

Hossam Afifi
INT-evry university, France
Email: Hossam.Afifi@int-evry.fr

***Abstract*— Bluetooth technology provides conveniences ranging from simply substituting for wires of electrical household products to constructing home network systems. Providing Bluetooth-technology-based services in public raises security issues in providing a secure Bluetooth link for unknown Bluetooth peers.**

In this paper, we present a bypassing security model for protecting communications between anonymous Bluetooth peers via wireless local area network (WLAN) and authentication authorization and accounting (AAA) technologies. Our bypassing security model is composed of the Bluetooth peer authentication, the Bluetooth key negotiation, and the link key generation. It brings a cost-effective realization in conjunction with standard technologies and is suitable for large-scale service providing systems by relying on the infrastructure network. In contrast to a certificate-based Diffie-Hellman method that requires computation-intensive cryptographic functions, our model performs faster on power-limited devices.

I. Introduction

Bluetooth technology provides great efficiency and cost savings for the home and business users, which also allows for cable replacement, ease of file sharing, wireless synchronization and Internet connectivity. Wireless e-wallet services are also attractive, i.e., an electronic charging and payment system in open areas like a subway station or parking lot.

Security concerns however arise in deploying e-payment services using Bluetooth technology in public. Since two Bluetooth devices that intend to communicate are virtually unknown to each other, authentication and privacy are utmost of importance. Bluetooth security is based on a key derivation scheme. A pair of Bluetooth devices are initially set to the same personal identification number (PIN) code. The two devices derive a link key from their PIN, and then an authentication and encryption keys are obtained in a consecutive derivation from the link key. This security mechanism is suitable for establishing a secure personal area network (PAN) among trust parties, but not among anonymous peers. Instead of exchanging the PIN code among unknown devices, Bluetooth security allows for importing a link key into the Bluetooth modules [1]. It nevertheless requires a secure key exchange method including authentication. We therefore focus on how to transfer a link key securely to a pair of anonymous Bluetooth peers, based on an infrastructure network.

In this paper, we propose a bypassing security model that allows for key negotiation via IEEE 802.11 Standard [2]. We develop three steps of a Bluetooth secure link establishment: peer authentication via EAP [3], security information exchange via AAA protocols [4], and a Bluetooth link key derivation among anonymous peers. The Bluetooth peer is pre-authenticated by its authentication server and security information, as a result of a successful authentication, is transferred to the authenticated peer and the Bluetooth-related AAA server, by means of Diameter-Bluetooth application that we also present in the paper. When the authenticated peer intends to associate with a targeted peer, two peers derive a Bluetooth link key based on the exchanged security information.

Our bypassing security model is compatible with current standard protocols, 802.11i authentication [5], that brings a cost-effective realization and suitable for a large-scale service providing model by relying on the infrastructure network. It provides a high-performance secure link establishment for power-limited portable devices, compared to a certificate-based Diffie-Hellman method that requires computation-intensive cryptographic functions.

The paper is organized as follows. Section II gives an overview of Bluetooth security and IEEE 802.11i authentication. Section III presents our security model in coordination with WLANs, AAA and Bluetooth. Section IV describes the procedure of Bluetooth security establishment that involves authentication, security information exchange, and build-up of a secure Bluetooth link. Section V shows the performance analysis of our security model via the simulation. After presenting related work in Section VI, we conclude this paper in Section VII.

II. Background

In this section, we explain basic Bluetooth security mechanisms and IEEE 802.11i authentication.

A. Bluetooth security

There are three modes of security for Bluetooth access between two devices.

- Security mode 1: no security
- Security mode 2: service level enforced security
- Security mode 3: link level enforced security

In security mode 1, security procedures are not initiated in a device. This mode allows any Bluetooth devices to connect to the others. In security mode 2, security procedures are initiated after a channel is established at the logical link control and adaptation protocol (L2CAP) level. In this mode, a security manager controls access to services and devices.

In security mode 3, security procedures are initiated before channel is established. This mode supports authentication and confidentiality, which is based on a secret link key shared by a pair of devices. The link key is generated while two Bluetooth devices associate with each other for the first time. Two associated devices simultaneously derive the link key from the PIN code. It is also possible to create a link key using upper layer key exchange methods and then import the link key into the Bluetooth modules.

Entities used in the authentication and encryption procedure are: 48-bit Bluetooth device address (BD_ADDR), 8 to 128-bit PIN, 128-bit link key for authentication, 32-bit authentication response ($SRES$), 96-bit authenticated cipher offset (ACO), 8 to 128-bit cipher key for encryption, and 128-bit random value ($RAND$).

The Bluetooth authentication process is based on a challenge-response scheme. Provided that one of the Bluetooth devices (the claimant) attempts to connect to the other (the verifier), the steps of the process are the following:

- The claimant sends its BD_ADDR to the verifier.
- The verifier challenges the claimant by sending $RAND$ for authentication.
- Both the verifier and claimant generate a $SRES$, by applying the Bluetooth E_1 algorithm with BD_ADDR, $RAND$ and the link key. The output results of E_1 are $SRES$ and ACO.
- The claimant returns its $SRES$ to the verifier.
- The verifier ensures if they are matched. The verifier, if so, continues security connection establishment.

The Bluetooth encryption process consists of cipher key generation and encryption. Provided that one of the Bluetooth devices is the master and the other is the slave, The steps of the process are the following:

- The master generates and sends $RAND$ for encryption to the slave.
- Both the slave and master generate an encryption key, by applying a key generator with $RAND$, ACO and the link key.
- The master sends its BD_ADDR to the slave.
- Both the slave and the master generate a key-stream, by applying the Bluetooth E_0 algorithm with the encryption key, $RAND$, BD_ADDR and real-time clock.
- Encryption and decryption are performed.

B. 802.11i authentication

802.11i authentication operates with three components: mobile station (STA), access point/authenticator (AP/AUTH) [1] and authentication server (AS). The STA and AP communicate with each other in WLANs. The AUTH and AS are connected through IP-based networks and protected with the help of security protocols like IPsec [6]. It consists mainly of 802.1X authentication, a 4-way handshake and, optionally, a 2-way handshake phases.

[1]AP communicates with STA and AUTH does with AS, yet AP and AUTH are tightly coupled in reality.

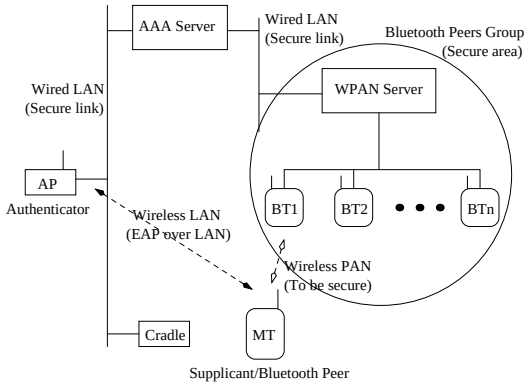

Fig. 1. The entities in a security framework combined with WLAN and WPAN

The 802.1X authentication phase involves the STA authentication and key distribution operations, based on AAA infrastructure [4]. After the success of the 802.1X authentication phase, three parties, namely the STA, the AP and AS hold the same pairwise master key (PMK). As the last of 802.11i authentication, the 4-way handshake phase is performed to generate a pairwise temporary key (PTK), shared between the STA and AP. Optionally, an additional group temporary key (GTK) is generated for a group of STAs.

III. Bypassing security model

Figure 1 illustrates the entities of our model. The description is the following:

- Mobile terminals (MTs) have the Bluetooth and 802.11 interfaces to internetwork connections.
- AAA server behind the APs is responsible for user authentication and Bluetooth key management. The link between the two is secure.
- Wireless PAN (WPAN) sever forms a group of Bluetooth terminals (BTs) and maintains Bluetooth keys with the BTs. On the other side, the WPAN server serves as an entity to exchange messages with the AAA server to generate a temporary Bluetooth key. The link between the two is secure.
- BT is a front-end Bluetooth link entity and constructs a secure link with the MT eventually. The BTs are within a secure area at the center of the WPAN server.

This security model defines three types of applications: EAP-Bluetooth application between the MT and AAA via the AUTH, Diameter-Bluetooth application between the AUTH and AAA server and between the AAA server and WPAN server, and the Bluetooth peer group management application between the WPAN server and BTs, all of which will be described in the following section.

This model is applicable for e-payment services in a subway station or parking lot; walking through a gateway, each of the MT and BT detects its signal via Service Discovery Protocol (SDP) [1]. a one-time key is inevitably to be generated automatically. The users may install a key in manual that is

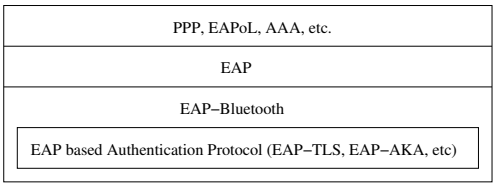

Fig. 2. EAP-Bluetooth of a protocol stack presented in IEEE 802.1X

provided in advance by a service provider associated with the subway station, which is not helpful to users.

IV. Bluetooth secure link establishment

In this section, we describe the Bluetooth link key derivation procedure, based on the bypassing security model.

A. EAP-based authentication

There are many EAP-based authentication protocols for WLANs, such as EAP-TLS [7]. Our security model does not specify any of EAP-typed protocols, but all are open to use. After the authentication process is completed, EAP-Bluetooth application begins to exchange the Bluetooth information.

B. Exchange of Bluetooth information

EAP-Bluetooth [8] is a carrier protocol for EAP-based authentication protocols. It provides a way to extend functionality of 802.1X authentication for application services. EAP-Bluetooth is used basically for Bluetooth key transfer application. Figure 2 illustrates a protocol stack including EAP-Bluetooth-carrying EAP-based protocols. EAP-Bluetooth does not define any authentication protocol, but allows additional operations to be performed after EAP-based authentication protocol is completed.

Upon receiving the EAP response message of Identity, EAP-Bluetooth starts with an EAP-based authentication protocol as we see in Figure 3. The Type field in EAP-Bluetooth is *Bluetooth* and the Data field contains the EAP-typed message transferred. Therefore, EAP-typed messages are carried over EAP-Bluetooth. Once the EAP-based protocol is completed, extensible operations for Bluetooth begins by sending the Bluetooth request message.

The Bluetooth request message contains supplicant's MAC address (BD_ADDR_X), a random value ($RAND$) and a destination name. The opposite Bluetooth device's MAC address is supposed to be transferred along with the message, but in case it is not available, it is omitted.

$$\text{Blue-Key req:} ID_X, BD\_ADDR_X, RAND_X, Dest\_Name$$

Upon receiving the request message, the AAA server responds with the message, including a newly generated random value.

$$\text{Blue-Key res:} ID_X, RAND_{AS}$$

In the meantime, it generates a temporary Bluetooth key with the two exchanged random values as follows:

$$\text{temporary Bluetooth key} = hash(RAND_X || RAND_{AS})$$

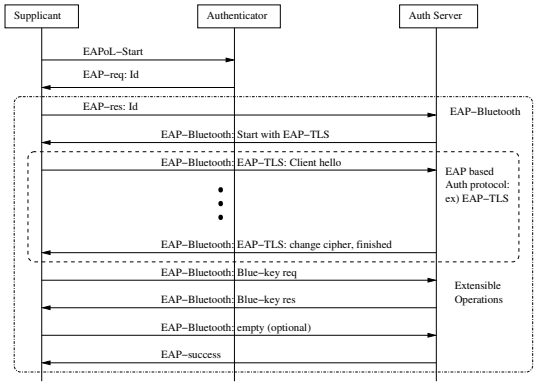

Fig. 3. Message exchanges of EAP-Bluetooth along with EAP-TLS in IEEE 802.1X

and sends the message including BD_ADDR_X and this key to the WPAN server. Accordingly, the supplicant can also generate the same key upon receiving $RAND_{AS}$ from the AAA server.

Diameter-Bluetooth application is the following. The command Codes for Bluetooth key messages in Diameter protocol [9] include `AAA-Bluetooth-Request (ABR)` and `AAA-Bluetooth-Answer (ABA)`.

The ABR message, indicated by the command-code field set to TBD (*To Be Defined*) and the 'R' bit set in the Command Flags field, is used for the MT (supplicant) to request a Bluetooth key from the AAA server. The ABR message format is the following:

```
AAA-Bluetooth-Request ::= < AVP header: TBD >
           < Session-Id >
           { Origin-Host }
           { Origin-Realm }
           { Destination-Realm }
           { Auth-Application-Id }
           { Bluetooth-Request-Type }
           [ B-Subscription-Id ]
          +[ BD-ADDR ]
           [ RAND ]
          *[ Destination ]
           [ User-Name ]
          *[ AVP ]
```

The ABA message, indicated by the command-code field set to TBD and the 'R' bit cleared in the Command Flags field, is used for the AAA server to respond to the ABR message and transfer a key to the WPAN server. The ABA message format is the following:

```
AAA-Bluetooth-Answer ::= < AVP header: TBD >
           < Session-Id >
           { Origin-Host }
           { Origin-Realm }
           { Destination-Realm }
           { Auth-Application-Id }
           { Bluetooth-Request-Type }
           [ B-Subscription-Id ]
          *[ BD-ADDR ]
          *[ RAND ]
          *[ Bluetooth-Key ]
           [ User-Name ]
          *[ AVP ]
```

Defined AVPs: Several AVPs (Attribute Value Pair) are defined for the AAA/Bluetooth application.
The `Bluetooth-Request-Type` AVP is of type Enumer-

ated and contains the reason for sending the AAA-Bluetooth request message. The following values are defined for the `Bluetooth-Request-Type` AVP:

`KEY_REQUEST` 1 : It is used to request the Bluetooth key from the AAA server. A pair of messages of request and response hold the same value.

`RAND_REQUEST` 2 : It is used to request a fresh random value to generate a key. A pair of messages of request and response hold the same value.

`ABORT_REQUEST` 3 : If a key negotiation fails, it is used to indicate to abnormally terminate the Bluetooth key request procedure. The message of the response to a key request holds this value.

The `B-Subscription-Id` AVP is of type OctetString and contains 16 octets of the Bluetooth query identifier. It is set to the Bluetooth MAC address of an initiator, such as MT. It is used as an identity for verifying a duplicating query for the same device.

The `BD-ADDR` AVP is of type OctetString and contains the 48-bit IEEE of Bluetooth device address unique for each Bluetooth unit.

The `RAND` AVP is of type OctetString and contains 16 octets with the 'P' bit enabled, which is used to generate the Bluetooth key as an element.

The `Destination` AVP is of type OctetString and contains 16 octets of a service provider identifier, i.e., the name of station/the identity of the WPAN server can be the one.

The `Bluetooth-Key` AVP is of type OctetString and contains maximum 16 octets with the 'P' bit enabled, which results from the generation of the Bluetooth key.

The other AVPs are used as defined in [9], [10].

C. Build-up of a secure Bluetooth link

Once receiving the message including BD_ADDR_X and the temporary Bluetooth key, the WPAN server handles it in an either centralized or distributed way to manipulate the key.

As shown in Figure 4-(a), when the MT(supplicant) approaches Bluetooth peers and detects Bluetooth signal from the one of BTs (BT_2 in the Figure), MAC addresses, $BD\_ADDR_{BT_2}$ and BD_ADDR_X, are exchanged. BT_2 forwards BD_ADDR_X received from the MT to request a link key from the WPAN server. The WPAN server then generates the link key by applying a pseudo random function with the parameters, BD_ADDR_X, $BD\_ADDR_{BT_2}$ and the temporary Bluetooth key received from the AAA server. The generated link key is sent to BT_2. In the meantime, the MT generates the same link key as BT_2.

Alternatively, using the distributed system in Figure 4-(b), the WPAN server first broadcasts the temporary Bluetooth key received from the AAA server to a group of BTs; each Bluetooth peer is allowed to generate a link key individually when the MT's Bluetooth signal is detected. As the MAC addresses of BT_2 and the MT are exchanged, the MT generates a link key by applying a pseudo-random function with the exchanged addresses and previously obtained random values. BT_2 uses the exchanged address and the temporary Bluetooth key.

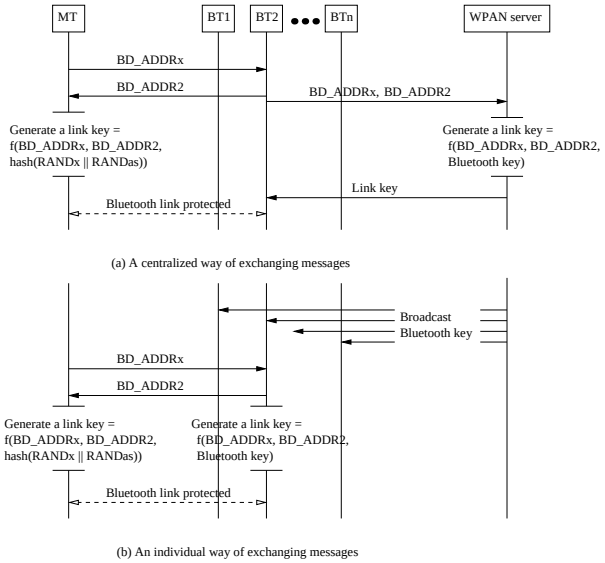

Fig. 4. Message exchanges between Bluetooth peers for a link key

Bluetooth authentication: Once the link key is derived on both BT_2 and the MT, the authentication process operates. The entity authentication in Bluetooth security uses a challenge-response scheme which is based on the knowledge of the link key. BT_2 authenticates the supplicant, following the procedure described in II-A.

Bluetooth encryption: If the authentication verification process is completed successfully, the communication information can be protected by encrypting the packet payload. The pair performs encryption and decription of the packet payload, following the procedure described in II-A.

V. Simulation

The goal of the simulation is to investigate the latency performance of our bypassing security model.

A. Simulation set-up

We conduct simulation based on experimental data. Table I shows the performance measurement of cryptographic functions. The experiment was carried out provided that the programs [11] are compiled on two different machines, Linux v.2.4.19 iPAQ 206MHz ARM processor with 64MB and Linux v.2.4.23 Intel Xeon 3GHz bi-processor with 2GB. In addition to these experimental data, we determine the constant value, extracted from [12] and [13], of the parameters used in the simulation in table II. The variation of traffic in wireless and

TABLE I

Throughput of symmetric and asymmetric algorithms

Algorithm\Power	iPAQ	Xeon
SHA-256	3.4 Mbps	64.0 Mbps
MD5	15.8 Mbps	290.9 Mbps
RSA encryption	15.1 Kbps	625 Kbps
RSA decryption	0.9 Kbps	21.6 Kbps
RSA signature	0.9 Kbps	21.2 Kbps
RSA verification	15.1 Kbps	625 Kbps

TABLE II

PARAMETERS USED IN SIMULATION

Parameters	Values
Probe time for WLAN	$37ms$
Open Auth.& Asso. time for WLAN	$4.6ms$
Bluetooth PAN setup time	$5sec$
RTT (local domain)	$2ms$

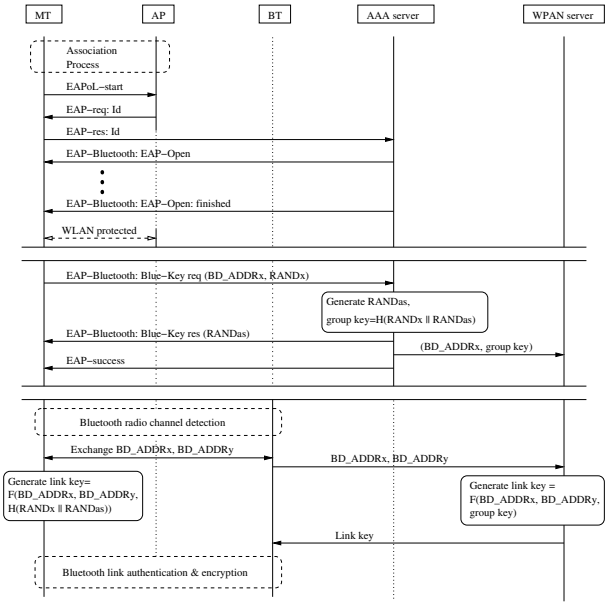

Fig. 5. Simulation scenario with message flows

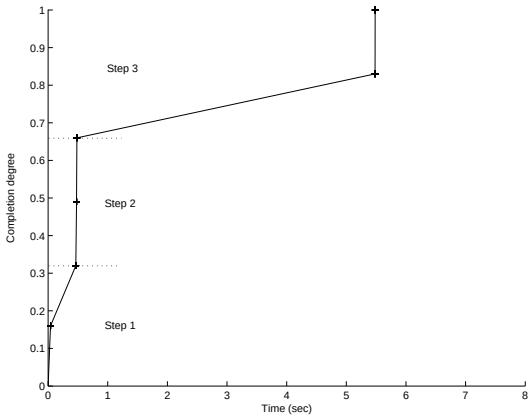

Fig. 6. The degree to the completion of processes required. No jitter is included.

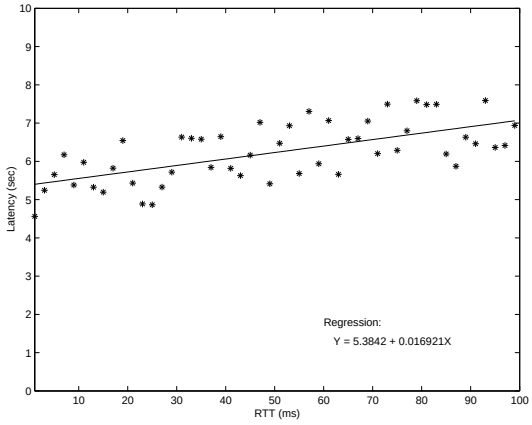

Fig. 7. The increasing degree to the total latency as RTT grows.

wired communications is less than 5% as default and the variation of latency to establish a Bluetooth link is regarded as less than 30%. The coverage ranges of all APs are identical circles. There is no delay caused by the interference from other MTs except the variations presented above.

The simulation scenario, shown in Figure 5, is partitioned into: Step 1) EAP-based mobile terminal authentication, Step 2) exchange of Bluetooth information, and Step 3) set-up of a link key. Note that the first two steps can be pre-computed apart from the last. MT and BT operate on iPAQ devices and the others do on Xeon machines. TLS is used to authenticate the MT in Step 1. AAA server keeps the same distance from AP and WPAN server, respectively. The establishment of a secure Bluetooth link in Step 3 is simulated in a centralized way of exchanging messages.

B. Simulation results

Step 3 is a dominating factor to degrade the latency performance: The result has been obtained from the measurement of completion degree by functionally partitioning the whole process into three steps. EAP/Bluetooth authentication process (Step 1 and 2), relying on the infrastructure, is completed less than 1 second as shown in Figure 6. In contrast, a key inquiry process (Step 3) that includes discovering Bluetooth devices to associate with incurs a considerable delay degrading the whole latency performance.

Adaptability to the inter-domain infrastructure: we clearly see in Figure 7 that our model is suitable for the inter-domain-based back-end authentication system. The increasing degree to the distance between the AUTH and AAA server, and between the AAA and WPAN servers has a minor impact on the entire latency.

Unstable Bluetooth association process: Figure 8 shows the variation of latency as the jitter of end-to-end links increases. We see the Bluetooth link is deeply affected by jitter, which means its delay incurs a significant amount of latency to the total latency performance; it is essential to improve the Bluetooth association functions.

Our model performs faster than the certificate-DH method: An alternative solution to exchanging a link key is the combination of certificate-based authentication and Diffie-Hellman (DH) key exchange. First, two peers exchange their certificate including encrypted DH public value. After verifying the received certificate successfully, each peer gets to obtain the public value by decryption; it leads to compute modulus operations and derive a secret key. Figure 9 shows the cumulative distribution function of latency required in securing Bluetooth links in both our model and certificate-DH method. In 6 seconds, more than 90% of secure link establishments are

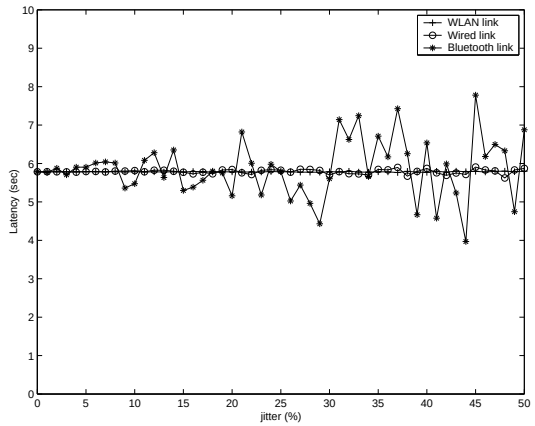

Fig. 8. Latency variation as to the increase of jitter.

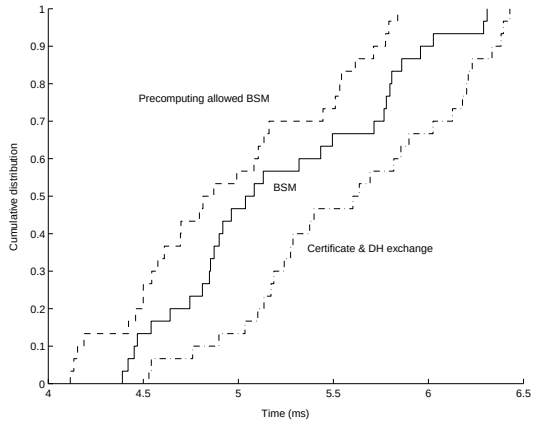

Fig. 9. The cumulative distribution of securing a Bluetooth link. Precomputing allowed bypassing security model (BSM) is enabled to perform Step 1 and 2 in advance somehow.

completed in our model. In particular, precomputing allowed BSM achieves 100% in less than 6 seconds, compared to around 70% in the certificate-DH method. The simulation result makes sure that when the MTs approach BTs from a 10 meter distance at $6km/h$ (an ordinary walking speed), our methods successfully provide secure links for all mobile terminals.

VI. Related Work

McCune *et al.* [14] presented a system for authentication and demonstrative identification of devices via a *visual channel* that is implemented with 2D barcodes and camera-phones. A barcode-typed certificate, transferred via the visual channel, precludes man-in-the-middle attacks among devices that share no prior context. Danzeisen *et al.* [15], [16] presented a hybrid architecture that consists of cellular-aware and non-cellular-aware mobile nodes. To secure communications between non-cellular nodes, secret information is transferred via a short messaging service (SMS) through cellular nodes. The latency of SMS message exchanges however takes about 37 seconds. Hoepman [17] focused on the ephemeral pairing problem in wireless ad-hoc networks and presented several ephemeral key exchange protocols. The protocols basically rely on Diffie-Hellman key exchange [18]. To avoid man-in-the-middle attacks, they assume that the communication channel is authentic. However, in reality the communication channels are not always authentic.

VII. Conclusion

Our bypassing security model for anonymous Bluetooth peers is appealing for this reason: it is highly compatible with wireless technologies standards so that implementation and deployment would be easily provided.

In this paper we develop the bypassing security model that enables anonymous Bluetooth peers to securely communicate with each other. It is configurable to allow precomputation for the tradeoff between the latency and availability.

The bypassing security model is faster than the certificate-based DH method, so it would perform faster on low-power Bluetooth devices because symmetric cryptographic functions provide an additional 2- to 3-fold speedup.

References

[1] B. SIG. (2001, Feb.) Specification of the Bluetooth system, Core Version 1.1. [Online]. Available: http://www.bluetooth.com/

[2] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Robust Security," in *ANSI/IEEE Std 802.11: 1999(E)*. ISO/IEC 8802-11, 1999.

[3] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "Extensible authentication protocol (eap)," RFC 3748, June 2004.

[4] D. Mitton *et al.*, "Authentication, Authorization, and Accounting: Protocol Evaluation," RFC 3127, June 2001.

[5] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Specification for Robust Security," in *IEEE Std 802.11i/D3.1*. ISO/IEC 8802-11, 2003.

[6] IP Security Protocol (ipsec). [Online]. Available: http://www.ietf.org/html.charters/ipsec-charter.html

[7] B. Aboba and D. Simon, "PPP EAP TLS Authentication Protocol," RFC 2716, Oct. 1999.

[8] H. Kim *et al.* (2004, Feb.) EAP Bluetooth Application. Internet draft. [Online]. Available: http://www.inria.fr/planete/hkim/downloads/papers/draft-kim-eap-bluetooth-00.txt

[9] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko, "Diameter Base Protocol," RFC 3588, Sep. 2003.

[10] P. Calhoun *et al.* (2003, Jun.) Diameter Network Access Server Application. Internet draft. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-aaa-diameter-nasreq-12.txt

[11] Crypto++ 5.1 http://www.eskimo.com/∼weidai/benchmarks.html.

[12] A. Mishra, M. Shin, and W. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," *ACM SIGCOMM: Computer Communications Review*, vol. 33, no. 2, pp. 93–102, 2003.

[13] M. Shin, A. Mishra, and W. Arbaugh, "Improving the Latency of 802.11 hand-offs using Neighbor Graphs," in *ACM Mobisys 2004*, Boston, Jun. 2004.

[14] J. M. McCune, A. Perrig, and M. K. Peiter, "Seeing-is-believing: Using camera phones for human-verifiable authentication," in *Symposium on Security and Privacy*. Oakland, USA: IEEE, May 2005.

[15] M. Danzeisen, T. Braun, D. Rodellar, and S. Winiker, "Heterogeneous network establishment assisted by cellular operators," in *IFIP TC6 International Conference on Mobile and Wireless Communications Networks (MWCN'03)*, Singapore, Oct. 2003.

[16] S. Winiker, "Integration of cellular assisted heterogeneous networking and bluetooth service discovery and protocol," Master's thesis, Bern University, 2004.

[17] J.-H. Hoepman, "The Ephemeral Pairing Problem," in *Financial Cryptography '04 (FC)*. Springer Verlag, Feb. 2004.

[18] W. Diffie and M. E. Hellman, "New Direction in Cryptography," 1976.