# Location Privacy in Bluetooth

Ford-Long Wong and Frank Stajano

University of Cambridge,
Computer Laboratory

**Abstract.** We discuss ways to enhance the location privacy of Bluetooth. The principal weakness of Bluetooth with respect to location privacy lies in its disclosure of a device's permanent identifier, which makes location tracking easy. Bluetooth's permanent identifier is often disclosed and it is also tightly integrated into lower layers of the Bluetooth stack, and hence susceptible to leakage. We survey known location privacy attacks against Bluetooth, generalize a lesser-known attack, and describe and quantify a more novel attack. The second of these attacks, which recovers a 28-bit identifier via the device's frequency hop pattern, requires just a few packets and is practicable. Based on a realistic usage scenario, we develop an enhanced privacy framework with stronger unlinkability, using protected stateful pseudonyms and simple primitives.

## 1 Introduction

### 1.1 Wireless Devices

Ubiquitous gadgets have been steadily proliferating, posing an increasing threat to personal information privacy. The types of ubiquitous devices may be simplistically arranged on a spectrum according to their intended pervasiveness. On one end we would have the personal cellphone, which we may find one apiece for each individual person, where each cellphone is identifiable and traceable by its network. At the other end, you may find by the hundreds RFIDs—passive radio tags returning 128-bit unique IDs. Privacy solutions for cellphones include the use of network-issued temporary pseudonyms - the 'TMSI', and ways to manage these [1, 2]. Solutions for RFID privacy include 'killing' the tag upon purchase of the attached item, or enclosing it in a mesh, or changing its ID by 're-encrypting' with an external agent, etc. We propose that short-range ad-hoc wireless technologies, such as Bluetooth, which lie in the middle of the spectrum among ubiquitous devices, lend themselves to a different solution framework. Bluetooth devices have finally appeared in large numbers in the past two years after initial problems, and have gained market acceptance and general user familarity. Improving upon a tested and well-received technology may be less painful than designing from ground-up a completely new solution. This article is an attempt to work towards a more refined privacy solution framework for Bluetooth.

R. Molva, G. Tsudik, and D. Westhoff (Eds.): ESAS 2005, LNCS 3813, pp. 176–188, 2005.

### 1.2 Location Privacy

There are different components to privacy. The Common Criteria [3] analyzes privacy into anonymity, pseudonymity, unlinkability and unobservability. Anonymity deals with whether a subject may use a resource without disclosing the user identity. Pseudonymity makes a user accountable for the use, without disclosing his identity, by providing an alias. Unlinkability ensures that a user may make multiple uses of resources or services without others being able to link these uses together. It attempts to obscure the relations between actions by the same user. Unobservability ensures that a user may use a resource without third parties being able to observe that it is being used. For example, a broadcast obscures from third parties who actually received and used that information.

In location privacy, we are concerned with a particular type of privacy, which has been defined as 'the ability to prevent other parties from learning one's current or past location' [4], and it is a relatively new issue in privacy.

### 1.3 Structure of Paper

In Sections 2 and 3, we cover attacks on the current system. Our location privacy goals are outlined in Section 4. Proposals to assure location privacy are described in Sections 5 and 6.

## 2 Vulnerabilities

Current well-known authentication weaknesses in Bluetooth could be relatively easily resolved by recourse to asymmetric key establishment techniques [5, 6] at the cost of slightly increased computation. These enhancements would defeat even a strong adversary, by which we mean one which is omnipresent, has significant computational resources, and is able to mount active attacks.

In comparison, it is generally difficult to secure privacy, including location privacy. Awareness to Bluetooth's vulnerabilities in this area was first raised by Jakobsson and Wetzel [7]. Each Bluetooth device is identified by a unique permanent 48-bit Bluetooth Device Address (BD_ADDR). As Bluetooth is usually attached onto personal devices, the detection of a particular BD_ADDR in the neighbourhood would suggest that a particular human operator is nearby. That individual may even be carrying multiple Bluetooth devices and, if such a cluster of BD_ADDRs is detected, it is highly probable that the individual is nearby.

Furthermore, the device's BD_ADDR is used as an input into many procedures in Bluetooth. It is deeply entangled into certain parts of the protocol stack, and it is difficult to engineer it away easily, showing the general difficulty of providing security as an afterthought.

We will provide an overview of aspects of the Bluetooth radio and baseband layers, which are cause for privacy concerns. They can be summarized into:

1. problems of discoverability
2. problems of the non-discoverable mode
3. disclosure of the identity in certain packets

4. derivation of the access code from the identity
5. derivation of the frequency hop set from the identity

We provide a survey of the first three problems, which are well-known. The last two problems have been raised before partially, but we analyze and quantify more fully the risks involved.

### 2.1 Problems of Discoverability

The purpose of discovery is to allow one to find devices which one has not encountered before. The inquiry scan mode is also known as discoverable mode. The discovering (or inquirying) device sends ID packets, which contain just an access code — either the General Inquiry Access Code (GIAC) or a Dedicated Inquiry Access Code (DIAC), and according to the requisite inquiry hop sequence. A device in the inquiry scan mode will respond to inquiries with a Frequency Hop Synchronisation (FHS) packet, disclosing its own BD_ADDR and CLKN (native clock). The response is not immediate, but is on receipt of the next packet, so as to avoid collision with other slaves.

Essentially, the discovery process enables a hitherto stranger device to be found after at most tens of seconds, and from the privacy perspective the real identity is unfortunately disclosed when a device is discoverable. Keeping devices constantly discoverable is clearly a privacy risk. It is advisable to turn off discoverability whenever it is not needed.

### 2.2 Problems of Non-discoverable Mode

Devices which are set to 'non-discoverable' are nevertheless responsive to some degree. If they are set to 'connectable', they can still be detected, due to privacy weaknesses in the page and page scan states. During page, the master device will page for another device using an ID packet containing a Device Access Code (DAC) derived from the Lower Address Part (LAP) of the latter's BD_ADDR. The hopping at the physical layer during page is similar to the case for inquiry. The hop sequence is derived from the DAC, instead of the GIAC or some DIAC, together with the estimated clock (CLKE) of the paged device. When the slave detects the page message containing its own DAC, it will reply with an ID packet containing the same DAC. After that, the master will transmit a FHS packet.

Thus, a slave device set to non-discoverable and connectable will not respond to inquiry messages, but it will respond to page messages containing its permanent DAC. Devices which have previously encountered this device and have a record of its BD_ADDR and/or DAC can still page for it successfully if the device is within radio range. If its BD_ADDR is not known, the discovering devices can conduct a brute-force search of the BD_ADDR range, or more precisely, the 24-bit LAP range. The only means of protection against being tracked this way possible under the current specification are to either turn off Bluetooth, or to switch to non-connectable mode if such fine-grained control is supported on the particular device, and lastly, to reduce occurrences of pairing to a minimum so as avoid over-exposing the device's BD_ADDR.

### 2.3 Disclosure of Identity in FHS Packets

The FHS is a special control packet. The entire BD_ADDR of the sender, comprising the Lower Address Part (LAP), Upper Address Part (UAP), and Non-significant Address Part (NAP), are disclosed in the FHS packet, together with the highest 26 bits of its 28-bit native clock CLKN. The FHS packet is sent on two occasions: by a slave device in inquiry scan mode responding to an inquiry; and by a master device in page mode responding in turn after a slave in page scan mode has responded to the page. The device's identity is hence revealed to the opposite party and to any eavesdropper who is monitoring the spectrum.

### 2.4 Baseband Access Code Derived from Identity

The derivation of the Channel Access Code (CAC) from the master device's LAP had been recognized by Jakobsson et al [7] as a privacy risk, because the LAP can be reverse-engineered. We generalize further that the derivation of not just the CAC but also the DAC from the LAP carry privacy risks.

The access code of a Bluetooth packet is either one of three types. The CAC is used during the Connected state, the DAC is used during page and page scan, and the Inquiry Access Code (IAC) is used during inquiry and inquiry scan. The sync word is a 64-bit code derived from the LAP of a BD_ADDR. In the CAC, the sync word is derived from the master device's LAP; in the DAC, the LAP of the paged slave unit is used; and in the IAC, either the single reserved LAP is used or else certain dedicated IACs are used. The inquiry state is of less interest for privacy because the IACs being correlated for are not too device-specific.

The attacker only needs to compute once a dictionary of $2^{24}$ (ie. 16.7 million) LAP entries and their corresponding 64-bit sync words. As raised in [7], when the attacker detects a CAC, he can perform a table lookup and learn the master device's LAP. For completeness, we further raise that when this attacker detects a DAC sent by a paging master and a responding slave, he can perform a table lookup using the same pre-computed dictionary, and learn the slave device's LAP. As such, the slave device, non-discoverable but connectable, also faces location privacy risks. Note that a particular LAP is not unique, though collisions would be rare. The remaining address bits — the 24 bits of the UAP and NAP which constitute the 'company_id', do not span the entire 24 bits of entropy — the allocated numbers are published by IEEE Standards, and as of Jan 2005, there were only around $2^{13}$ issued numbers.

### 2.5 Hop-Set Derived from Identity

Jakobsson et al [7] observed that since the hop sequence in a connected piconet is a function of the master device's BD_ADDR and CLKN, and thus if one can capture a FHS packet sent by the master, the hop sequence can be trivially calculated. We investigated the reverse attack—the more difficult one of how to

recover the master device's address by tracking the frequency hopping pattern if we failed to capture the master's FHS packet, and we found that collecting just 6 packets, along with other information, is adequate. This attack produces 28 bits of address — 4 more bits than attacking the access code.

Bluetooth uses frequency-hopping mainly to mitigate environmental interference, and to reduce collisions among different piconets. There are five types of hopping sequences for the 79-hop system, one type each for the inquiry, inquiry response, page, page response and connected states. Each of these sequences is determined by the 24-bit LAP and the lower 4 bits of the UAP, of the relevant device's BD_ADDR, and its clock. The choice of device address used here is identical to that used to compute the access codes for the different states.

Thus 28 bits of LAP/UAP and 27 bits of the clock go into the hop selection box at any one time to choose one frequency. This function is fully documented in the specification and is strictly surjective. In the connected state, the output selects one of 79 frequencies, corresponding to an approximate 6-bit range. Based on a reasonable assumption of a uniform distribution, thus for the same clock offset, roughly $2^{22}$ LAP/UAP values would result in the same frequency.

We can carry out the following attack. Capture a first packet and form a tuple of the clock and frequency. Do a brute-force search and narrow the set of $2^{28}$ LAP/UAP values into a set of about $2^{22}$ possibilities. Collect another packet and obtain the tuple. Assuming uniform distribution, we can narrow further to a set of $2^{16}$ possibilities. Continuing in a similar way, just 6 packets in total are required to determine a unique 28-bit LAP/UAP with a probability calculated at 99+%. This is described in Appendix A. The overall work factor is on the order of $2^{28}$. With so few packets required for successful attack, the attacker may simply listen at a fixed frequency for it to be re-visited, instead of scanning the entire band. We have to add a caveat that, since the clock setting at each packet is required, determining the master device's clock setting initially without recourse to capturing its FHS packet would entail an indirect route of obtaining a LMP packet containing the slave's clock offset relative to the master's, and inquiring the slave (which needs to be discoverable) to learn the slave's clock.

This novel attack shows that even if a master device is non-discoverable and non-connectable, its hop pattern in a connected state and a discoverable slave could betray its identity.

Bluetooth was not expressly designed to be resistant to interception and deliberate narrowband jamming, unlike, for example, military tactical communications. Our interest with the frequency hop in Bluetooth is on the anonymity issues rather than availability. By resource-sharing the radio access via different clock offsets and public long-term identifiers, frequency hopping achieves equitable allocation of the spectrum and reduces collisions, but it hurts privacy. To improve privacy, the options are: either to disentangle the identifier from the time-frequency allocation, thereby requiring a re-design of the radio layer; or else to just de-link the identifier from the long-term identity, which is simpler.

## 3 Adversary Types

We identify two classes of adversaries, in ascending order of capability to compromise the privacy of the Bluetooth device.

The first class of attackers use commercial Bluetooth devices that can inquire and page as usual, and can therefore find any discoverable Bluetooth devices, as described in Section 2.1. The attacking range may be extended by directional antennae, a concept well-known to EM/RF engineers. For example, a 18 dBi yagi antenna can boost the 100 m Class I Bluetooth range to around 900 m, and a 24 dBi antenna to 1.6 km, assuming low RF losses at the joints, though such antennae are large and obtrusive. Within this class of attackers, we can distinguish a slightly more sophisticated sub-class, who can conduct brute-force searches of the BD_ADDR space, or rather, the LAP space, so as to find connectable victim devices, as described in Section in 2.2. Such proof-of-concept code has been released [8], though it is estimated to require around 11 hours to conduct a complete search of the space, using 127 devices working in parallel. We have developed our own version of this attack using a shell script, the open-source BlueZ stack, and an ordinary Bluetooth dongle.

A second class of attackers uses radio receivers, or modified Bluetooth devices, which are not constrained to frequency hop. The first sub-class can listen on one selected channel continuously for all types of messages in the inquiry, inquiry scan, page, page scan, and connected state hops. If this attacker sees a CAC or DAC, he can carry out his table lookup privacy attack, as described in Section 2.4. If he sees a FHS packet, then he has learnt the full BD_ADDR, as described in Section 2.3. He can also derive the master's identity by knowing at which clock offsets a particular hop frequency is re-visited, and by probing a discoverable slave, as described in Section 2.5. Another more powerful sub-class is capable of listening on the entire 2.4 GHz band simultaneously. This attacker is less likely to miss any packets, and is more effective than the first sub-class in determining the CLKN of the target master device for the attack in Section 2.5. Attacking the access code is less costly than attacking the frequency hop pattern though. The first sub-class of attacks can be readily demonstrated with today's Bluetooth protocol analyzers, such as the Frontline-Tektronix BPA-100 and 105.

We distinguish between hardware, and do not distinguish between the cryptographic capability among the classes, because programs which do such computations can be commoditised easily and can run on generic PCs. The first category of adversaries are able to successfully compromise the privacy of today's Bluetooth devices easily, unless tight discipline is maintained over the use of the discoverable mode and connectable mode. The second category of attackers is able to compromise the privacy of Bluetooth devices even when their victims maintain tighter discipline over discoverability and connectability, and whenever devices are transmitting in a connected state. The overall efficacy of location privacy attacks also depends on the pervasiveness (and investment) of the attackers, and how effectively they can correlate and fuse information obtained by their various spatially distributed sensors to continuously track the location of their victims.

## 4 Location Privacy Goals

The current specification of Bluetooth does not support strong location privacy. Before we go into the detailed technical mechanisms, we need to define the usage scenarios for this short-range wireless connectivity technology. Then we will articulate the privacy goals which take into account the usage.

Bluetooth-equipped devices tend to talk to other personal devices, and less with fixed immobile network infrastructure. The interaction is mostly peer-to-peer. Users of Bluetooth do not seem to require it to have substantial location-awareness for it to work well for cable-replacement. A higher application layer may require location-awareness, but Bluetooth, as a connectivity layer, does not require location-awareness built-in, and can very well lean towards the location-private part of the continuum. These differences make its location privacy requirement different from other technologies which have been analyzed elsewhere, which had assumed a network backbone [4]. We admit that the security interaction of Bluetooth with the location-aware parts, where present, of the host device may merit further study.

On the other hand, devices hosting Bluetooth are rather much smarter than dumb tags such as RFIDs. Bluetooth interactions may be stateful, since session keys need to be established. Identifiers are required for this and cannot be eliminated. This is true for both piconet and scatternet configurations.

Temporary throwaway pseudonyms [9, 5, 10] can be of help. However, these must not be completely stateless, otherwise prior pair-wise relationships and piconet configurations would be quickly lost, and require frequent re-initialization. From the point of view of privacy, the need for a permanent identifier is debatable. Apart from helping manufacturers tell their product lines apart, having hierarchically arranged BD_ADDRs does not appear to do privacy much good.

Spectrum allocation and collision avoidance at the physical layer have been mentioned to have privacy implications. A good solution must resolve these.

While we have discussed exclusively about Bluetooth, in practice some other protocol is sometimes tunneled over Bluetooth. One important issue for anonymity is that the different protocols must carry out proper de-identification between them and be stateless. For example, if TCP/IP is tunneled over Bluetooth, the BD_ADDR should be de-linked from the IP address. However, we will consider this as outside the scope of this article.

Thus we require a privacy framework which provides sender and destination anonymity in a mostly peer-to-peer ad-hoc wireless environment. Pseudonyms may be used, and unlinkability between pseudonyms should be provided. The solution should account for cases in which the wireless personal area network stays in a static configuration, and for cases where state needs to be kept between two paired devices over different sessions due to the inconvenience of establishing a new session key. Unobservability should be provided. If the premises underlying the usage scenario evolve, the privacy framework needs to change too. The means to establish strong pair-wise keys is assumed to exist [5, 6]: this is a non-goal.

## 5 Problems of Pseudonyms and Permanent Identifiers

As the identifier BD_ADDR is tightly integrated in the protocol and is used in many computations, it cannot be easily discarded. Throwaway pseudonymous 'BD_addr_actives' were proposed by Gehrmann and Nyberg [5] to be used within an anonymity mode. Using frequently changing pseudonyms would improve the unlinkability between actions by the same actual principal, and also protect the permanent BD_ADDR, which the device still retains, from disclosure to a casual observer. Using pseudonymous BD_addr_actives this way also allow the original design of the access code and frequency hop to be essentially retained.

However, that proposal has three privacy weakness. The first is that the real identity, the BD_ADDR, is being used and may be disclosed to any device with which one has paired previously, though the identity is protected against other casual observers. Thus, adversaries can link different actions to the same actual principal if they can pair with this device, no matter what its particular BD_addr_active is at the instant. This is not an ideal privacy quality to possess, as policy-wise it should not automatically be assumed that all devices which have paired with one's own are not adversarial with respect to one's privacy.

A second weakness is with regards to the usage of BD_addr_alias, which is another 'BD_ADDR-like' identifier, established by two devices after they have paired, to signify the pairing in their respective database. For example, a BD_addr_alias would in Alice's database serve as an alias signifying Bob to Alice, and in Bob's database as an alias signifying Alice to Bob. In Alice's database there would be a tuple containing this BD_addr_alias and Bob's real BD_ADDR. In one mode, after Alice pages Bob, and before authentication takes place, Alice would send a packet containing this BD_addr_alias to Bob in an attempt to find out if they have paired before. Bob will now look up this alias in his database to find Alice's BD_addr, and respond accordingly. The problem with this usage is as follows: if Alice pages for Bob, but this is intercepted by an adversary Eve, and Eve receives the BD_addr_alias sent by Alice, while Eve will fail the test, Eve would be able to page Bob later using the alias, and thence be able to probe whether Bob has previously paired with Alice. The observability of transactions between Alice and Bob could thus be compromised offline.

A third privacy weakness is related to the second. An adversary who observes the same pairwise BD_addr_alias transmitted can deduce that the same two devices may be communicating again. There are other caveats concerned with the use of temporary pseudonyms, which we would discuss. One of the most germane ones is that if a device could continually be tracked, even as it changes its pseudonym, that could still be linked to the previous one.

We propose an enhanced anonymity mode, also using pseudonyms, which would attempt to address these three said problems, while recognizing that pairings may be stateful. We emphasize that this mode by itself will not resolve all privacy risks; a policy which requires discoverability and connectability to be turned off most of the time must be applied.

## 6 Proposed Solution: Protected Stateful Pseudonyms

### 6.1 Inquiry and Inquiry Scan

For device discovery, we keep to the Gehrmann and Nyberg proposal [5], where the inquiry and inquiry scan states are left as according to the original specification, with the change that the identifier returned at inquiry scan is the slave's BD_addr_active instead of its BD_ADDR.

As a matter of strong privacy policy to counter tracking, we recommend that a device's discoverability should be turned off whenever it is not required.

### 6.2 Page and Page Scan

The Gehrmann and Nyberg proposal featured two paging situations. One situation is where a master pages a slave based on the latter's current BD_ADDR_active. The second situation is where a master pages a slave based on the latter's long-term BD_ADDR, which is useful for previously paired units. The second situation allows pairings to be remembered, but has the unfortunate weakness of leaking the BD_ADDR of the slave being paged, hence compromising linkability as well.

We prefer that the long-term BD_ADDR never be leaked. We hence propose a somewhat different second situation, in which a master would attempt to page a slave using modified ID packets derived from the previous BD_ADDR_actives which the master and the slave had used to pair. These packets cryptographically protect the addresses from casual sniffing. The formats of the packets and the required protocol are as described in the following section. We believe that it is more private to have done pairing with the pseudonyms than with the long-term identifiers. This is not too difficult to support, as Bluetooth pairing is already based on a shared password rather than on permanent identifiers. It can be decided by policy settings how soon to expire pairings, as well as how soon a device expects a paired device to have changed pseudonyms.

As a policy setting, we recommend that a device's connectability be turned off whenever the owner does not expect connection requests to be received.

### 6.3 Protected Pseudonyms

This protocol (Fig. 1), designed for our second situation described above, attempts to protect past pseudonyms from all third parties. We modify the ID packet from the original Bluetooth specification. We now use three ID packets, denoted by ID1, ID2 and ID3. The relevant past pseudonyms of Alice and Bob are denoted by $I_A$ and $I_B$. $H$ is a hash function, $R_1$, $R_2$ and $R_3$ are random nonces, and $K_{AB}$ is the shared link key formed by Alice and Bob previously. The three-way handshake is essential. Say, Alice intends to page for Bob. On verifying correctly the ID2 packet, Alice will have the assurance that Bob knows his previous pseudonym, her previous pseudonym, and their shared key. On verifying correctly the ID3 packet, Bob has the assurance that Alice knows these same three things.

| # | Alice | | Bob |
|---|---|---|---|
| 1 | Chooses random $R_1$ | | |
| 2 | $H_1 = H(I_B \mid R_1 \mid K_{AB})$ | | |
| 3 | | $-\ ID1 : (R_1 \mid H_1) \rightarrow$ | |
| 4 | | | Verifies $H_1$ |
| 5 | | | Chooses random $R_2$ |
| 6 | | | $H_2 = H(I_A \mid R_1 \mid R_2 \mid K_{AB})$ |
| 7 | | $\leftarrow\ ID2 : (R_2 \mid H_2) -$ | |
| 8 | Verifies $H_2$ | | |
| 9 | Chooses random $R_3$ | | |
| 10 | $H_3 = H(I_B \mid I_A \mid R_1 \mid R_2 \mid R_3 \mid K_{AB})$ | | |
| 11 | | $-\ ID3 : (R_3 \mid H_3) \rightarrow$ | |
| 12 | | | Verifies $H_3$ |

**Fig. 1.** Protected Stateful Pseudonyms

Alice keeps a database of tuples each containing her temporary pseudonym, the pseudonym of the other party, and the shared link key. Bob keeps a similar database. Alice wants to page for Bob. She selects a random nonce $R_1$, computes the hash $H_1$, and sends an ID1 packet. The hash in the ID1 packet hides the past pseudonym of Bob. Bob would compute and verify the expected hash in the ID1 packet using his list of the paired devices' pseudonyms and their associated link keys with the nonce. When he successfully finds a match, he chooses a random nonce $R_2$, computes $H_2$, and responds with the ID2 packet. The hashes are inexpensive operations, thus the parties can do these easily. As Bob generates nonce $R_2$ randomly, he can be sure that his challenge to Alice is fresh. Alice, on receiving the ID2 packet, will verify the hash. If there is a match, Alice will generate a nonce $R_3$, compute the hash $H_3$, and reply with the ID3 packet. Bob will verify the hash on receipt of the $ID3$ packet. After the protocol runs successfully, both parties can proceed to carry out mutual authentication as usual. The security of the protocol depends on the randomness of the nonces, the irreversibility of the hash function, and the secrecy of the shared link key.

A naive replay attack — the second weakness mentioned in Section 5 — incarnated here as an adversary capturing an ID1 packet previously sent by Alice and received by Bob, and replaying it, would be defeated, because Bob checks for freshness of $R_1$ and $R_3$, and Alice checks for freshness of $R_2$

In another conceivable and more sophisticated attack, an adversary Eve intercepts an ID1 packet and prevents it from reaching Bob, but replays it later to Bob. Such an ID1 packet will pass Bob's $R_1$ freshness test. However, Bob now sends an ID2 packet with a fresh $R_2$. We can set a policy whereby uncompleted handshakes would raise an alarm at Bob's end, to alert Bob of the possibility of an intruder, so unless Eve next responds with a correctly formed ID3 packet, Bob would receive an alert. The 3-way handshake is essential for mitigating such an attack. Over at Eve's end, on her receipt of Bob's ID2 packet, Eve may suspect that Alice and Bob had paired previously, but she retains some doubt, because of possible collisions among the hashes.

The protocol is not resistant to an online relay attack — in which Eve would position herself between two widely geographically separated victims — because the protocol does not incorporate any distance-bounding algorithm.

We leave it open whether the length of the ID1, ID2 and ID3 packets need to be equivalent to the DAC length of 68 bits. If they are also 68-bit, especially the ID1 packet, then it helps to obscure the fact from simplistic traffic analysis that Alice is paging for a old pseudonym of Bob, in which case the random nonce would take up, say, 34 bits, and the hash the other 34 bits. Or else these packets can extend up to the length of 160 bits plus a suitable length of a nonce.

The proposed protocol provides good scalability in remembering and responding to past pairings, while not leaking the permanent BD_ADDR nor previous pseudonyms unnecessarily. Bob keeps changing pseudonyms, yet remains able to respond to some previous pseudonym of his which Alice has paired with, as he has kept a history of his pseudonyms, whose ages are set by policy.

### 6.4 Physical Layer

We have described in Section 2 that parts of the device address can be recovered from the access code and the frequency hop pattern. This privacy risk can be resolved by using changing pseudonyms. A more complex possible solution would be to modify the physical layer. The frequency hop pattern can, for example, be initialized from other parameters instead of a device's identifier and its clock. Another alternative solution is to use direct-sequence (DS) spread spectrum instead of frequency hopping, so that different DS sequences use different pseudonymous identifiers. But the cost of DS is generally considered higher.

### 6.5 Triggers for Pseudonym Change

We propose several triggering mechanisms to change pseudonyms. It is well-known that if a device can be continuously tracked, such as when it is discoverable and is the only device in a locality, then even a change of identifier would not prevent linkability. Discoverability ought to be turned off during pseudonym change. We suggest a sub-state in the anonymity mode in which the device is ready to change pseudonyms. A change may be triggered by any of several events. Firstly, it may be brought about by the owner's manual action. Secondly, it can be automatically changed at random time intervals. Thirdly, the pseudonym be changed when a certain threshold large number of discoverable devices are detected in an inquiry sweep. The rationale is that it would be easy to 'blend in with the crowd' and anonymise oneself. This method should be carefully applied because an attacker can spoof the presence of a large number of devices. [1] It uses the concept of a mix zone [4], the difference being that here, pseudonym change is handled by the devices themselves instead of a network infrastructure.

[1] However, the attacker would not reduce the anonymity of the victim by forcing a pseudonym change — the only effect would be to make the victim believe he is more anonymous that he actually is, which might perhaps lead him to lower his guard.

### 6.6 Further Issues

New pseudonyms must be randomly generated, and one solution is by hashing some counter. Also, the 28-bit 3.2 kHz Bluetooth device clock, which has a cycle of 23.3 hours, of which the highest 26 bits are disclosed (— or a 1.25 ms resolution), must be randomly re-adjusted on a pseudonym change, to prevent an adversary from linking pseudonyms to a clock, even accounting for the clock drift. Bluetooth uses a 'friendly name', which is a human-readable name to tag devices during device discovery and to help manage the list of paired devices locally. To reconcile privacy with usability, we propose the following: the field should be left empty or not transmitted during device discovery, but the user could be allowed to locally tag his list of paired devices with 'friendly names' of his choice to help him better distinguish the devices than through hexadecimals.

Certain RF attacks attempt to pinpoint the location of devices by measurements of irradiated power, and more sophisticated attacks distinguish RF signatures of individual devices, but these are outside our scope. In our privacy framework, we have not made use of digital certificates, because though these allow strong authentication, they are inimical to anonymity.

## 7 Conclusions

We have investigated the privacy problems of this pervasive wireless ad-hoc technology, particularly the leakage of its unique device address. We have surveyed known attacks against its location privacy, expanded a previously raised attack, and quantified another less-studied attack. The last attack requires only several packets and a work factor of $2^{28}$. While the basic location privacy problem of using a long-term device address can be resolved by using temporary pseudonyms, an incomplete solution can give rise to linkability.

Based on a plausible usage scenario distinct from other wireless technologies, we propose ways which refine the use of the pseudonyms, so that they are stateful and past device pairings can be remembered according to policy, yet which do not leak past pseudonyms and the long-term device address unnecessarily. We have also described various mechanisms to manage pseudonym change.

## Acknowledgement

We are grateful to the anonymous reviewers for their helpful comments.

## References

1. D. Kesdogan, H. Federrath, A. Jerichow and A. Pfitzmann. "Location Management Strategies increasing Privacy in Mobile Communication Systems". *Proceedings of the 12th IFIP SEC*, 1996.
2. S. Capkun, J. Hubaux and M. Jakobsson. "Secure and Privacy-Preserving Communication in Hybrid Ad Hoc Networks". *EPFL-IC Technical report IC/2004/10*, Jan 2004.

3. ISO/IEC-15408 (1999). *ISO/IEC-15408 Common Criteria for Information Technology Security Evaluation v2.1*, 1999. `http://csrc.nist.gov/cc`.
4. A. R. Beresford and F. Stajano. "Location privacy in pervasive computing". *IEEE Pervasive Computing*, **3**(1):46–55, 2003.
5. C. Gehrmann and K. Nyberg. "Enhancements to Bluetooth Baseband Security". *Proceedings of Nordsec 2001*, Nov 2001.
6. F.-L. Wong, F. Stajano and J. Clulow. "Repairing the Bluetooth Pairing Protocol". *Thirteenth International Workshop in Security Protocols*, Apr 2005.
7. M. Jakobsson and S. Wetzel. "Security Weaknesses in Bluetooth". *Proceedings of the RSA Conference*, **LNCS 2020**, 2001.
8. O. Whitehouse. "RedFang". 2003. `http://www.atstake.com/`.
9. D. Chaum. "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms". *Communications of the ACM*, **24**(2):84–88, Feb 1981.
10. M. Gruteser and D. Grunwald. "Enhancing Location Privacy in Wireless LAN through Disposable Interface Identifiers: A Quantitative Analysis". *First ACM International Workshop on Wireless Mobile Applications and Services on WLAN Hotspots*, 2003.
11. Bluetooth SIG Security Experts Group. *Bluetooth Security White Paper*, **1.0**, April 2002.
12. Bluetooth Special Interest Group. "Bluetooth Specification Volume 1 Part B Baseband Specification". *Specifications of the Bluetooth System*, **1.1**, Feb 2001.
13. Bluetooth Special Interest Group. "Bluetooth Specification Volume 2 Part H Security Specification". *Specification of the Bluetooth System*, **1.2**, Nov 2003.

## Appendix A: Recovery of Address Bits from Frequency Hop

The mathematics of the method can be formulated as a binomial distribution. We assume that each of the 79 outputs is equi-probable. We want to find the probability that after $k$ rounds, only one input is left, ie. all of the other $2^{28} - 1$ inputs are discarded at some round. Each one of these remains with probability $(1/79)^k$. Assuming independence of clock values, and independence between the outcomes of different inputs, the probability we seek is

$$(1 + x)^n = (1 - (\frac{1}{79})^k)^{2^{28}-1}$$

As the exponent is large, the numerical result is difficult to compute. Since $x$ is small with respect to 1, we can do a binomial expansion.

$$(1 + x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots$$

For $k = 6$, the first two terms sum to 0.9989. If we approximate $1/79$ to $1/2^6$, the result is 0.9961.