

Enabling Secure Ad-hoc Group Collaboration over Bluetooth Scatternets

Somil Asthana

Department of Computer Science
State University of New York
Buffalo, NY 14226
asthana@cse.buffalo.edu

Dimitris N. Kalofonos

Nokia Research Center
5 Wayside Road
Burlington, MA 01803
dimitris.kalofonos@nokia.com

Abstract— This paper presents an application-driven framework to enable secure ad-hoc group collaboration using Bluetooth (BTH). We do not intend to modify the BTH specified security; instead, we propose a new scatternet topology formation protocol designed with the BTH link-level security mechanism in mind. Our protocol enables group collaboration scenarios, where individuals participate in the scatternet with their private piconets, while maintaining existing sessions and security associations among their devices. Our protocol leads to a loop-free, compact tree topology, which allows for simple routing using existing Personal Area Network (PAN) profile functionality, without the use of additional ad-hoc routing protocols. We developed a prototype implementation of this framework and we provide initial experimental and simulation results that demonstrate the feasibility of our approach using actual BTH hardware.

I. INTRODUCTION

Bluetooth (BTH) [1] is currently the leading wireless technology for short-range Personal Area Network (PAN). The appeal of using BTH stems from its main design characteristics: *low cost*, with low complexity design allowing for low-cost hardware; *low power*, with efficient power saving modes and radio for low power consumption; *security*, with effective link-level authentication and encryption mechanisms, which are fairly easy to setup and use; *QoS Support*, with support for both synchronous and asynchronous data transmission; *co-existence capabilities*, with frequency-hopping allowing it to withstand interference in the unlicensed 2.4 GHz band; and, finally, *ad-hoc connectivity* with low-level device and service discovery mechanisms which make it possible to find and connect to previously unknown devices.

BTH devices discover other devices by following a resource intensive process called *inquiry*, during which the devices cannot perform any communication. On the other hand, devices that want to be discovered perform a much more light-weight process called *inquiry scanning*, during which they can continue to communicate. After the discovery phase, two devices choosing to connect follow a shorter process called *paging*, after which one de-

vice (*master*) ends up in control of the link to the other device (*slave*) and the simplest BTH network, a *piconet*, is created. Devices may also agree to perform a master-slave role switch. A master can have up to seven slaves in its piconet. To allow for the creation of a larger network called *scatternet*, devices participate in more than one piconets and provide bridging by assuming the composite role of master/slave or slave/slave.

The BTH standard specifies an effective link-level security mechanism for piconets. It uses the SAFER+ block cipher algorithm for link key and encryption key generation [2]. The link key is computed from the user's PIN at connection time and is used to authenticate the remote device. On successful authentication, this link key is used to produce the encryption key, for data encryption and decryption. Finally, instead of a piconet-wide common key, it is possible to use a different key in each master-slave link for increased security.

Because the BTH standard does not specify how scatternets are to be created, this topic has recently attracted an intense research interest. However, most of the existing proposals focus mainly on the topology formation aspects, assuming users' devices are nodes that can connect arbitrarily, without considering existing security associations and active sessions. Furthermore, little attention is paid to the applications that such networking protocols should support. We believe that when designing a BTH network formation protocol, it is important to consider the application area and usage scenarios in order to derive the requirements. Depending on the application area some approaches may be less appropriate than others or even not applicable at all.

In this paper we present an application-driven framework to enable secure ad-hoc group collaboration using BTH. We do not intend to modify the BTH specified security; instead, we propose a new scatternet topology formation protocol designed with the BTH link-level security mechanism in mind.¹ Our topology formation

¹As an aside, we also argue that in applications with very high

protocol allows for fast creation of fully connected scatternets, without requiring all nodes to be within radio coverage of each other. The resulting topology is a tree which, despite its shortcomings, is a loop-free topology. This allows for the creation of an Ethernet and IP local-link on top of a scatternet just by using standard PAN profile [3] functionality, without the need for any ad-hoc forwarding protocol. Finally, in this paper we describe our prototype implementation and provide some initial experimental and simulation results.

The remainder of this paper is organized as follows: Section II reviews existing related scatternet proposals; Section III gives our motivation and design goals; Section IV describes our scatternet formation protocol; Section V presents performance results; finally, Section VI presents our conclusions and future directions.

II. RELATED WORK

Scatternet protocols can be categorized based on the resulting topology, which can be a tree topology (e.g. [4], [5]), a mesh topology (e.g. [6], [7]) or some variant of mesh like a ring topology [8]. Also, some protocols can be categorized as static (e.g. [4], [6]), while others support dynamic network formation (e.g. [5], [9]) where the devices can arrive or leave arbitrarily. Finally, protocols can be characterized as centralized (e.g. [6]) or decentralized (e.g. [4], [5]), depending on whether they require nodes with special roles or not.

A mesh topology is more robust than a tree topology and avoids bottlenecks, but it incurs a substantial routing and inter-piconet scheduling overhead. Therefore, in our approach we chose a tree topology. BlueTree [4] is a static distributed tree topology protocol which requires each node to have prior knowledge of its neighbors. Tree Scatternet Formation (TSF) [5] is a dynamic distributed tree topology protocol which allows nodes to join or leave arbitrarily. A similar approach was followed in [9], which added features to avoid loops and help healing. Both [5], [9] form scatternets incrementally, by forming increasingly larger subtrees and trying to merge them. They also use the same initial discovery process where nodes alternate between inquiry and scanning. However, they require the frequent establishment of temporary connections to exchange information, which can add a significant overhead in actual networks.

Although the above proposals provide different solutions to the problem of scatternet formation and maintenance, they do not consider the impact of security in the protocols. To the best of our knowledge only [10] considers the issue of security in scatternets. However, the authors consider only the following two cases: (a) scatternets that are formed by individual nodes without ex-

isting security associations that create secure links based on a common user PIN; (b) “private PAN’s” with existing security associations that connect to an already formed insecure scatternet. In both cases they use a scatternet formation protocol that creates mesh networks and that requires the introduction of new Link Manager Protocol (LMP) commands, beyond the BTH standard specification [1].

III. MOTIVATION AND DESIGN GOALS

A. Motivating User Scenario

In this section, we illustrate a typical group collaboration scenario and establish the requirements for our scatternet formation protocol:

John decides to organize a meeting with his teammates Kia, Frank, Mary and Linda to complete their student project. He sends an SMS message and invites them to meet in the science library. Everyone, except Mary, receives the message immediately and comes to the science library with their BTH-enabled devices connected in private piconets. Frank comes with his laptop paired to his mobile phone and starts synchronizing them while waiting for the others. Linda comes listening to an mp3 song on her paired BTH headset. John decides at some point to initiate the meeting using his laptop and passes a paper with the meeting name and password to the rest. All enter the password in their laptops and join the meeting. Frank continues his synchronization session and performs a back-up of his phone files, while Linda continues listening to her mp3 song for a couple of more minutes. Soon they are all connected and start exchanging files related to their project. They can access Frank’s project files stored in his laptop, but he does not allow access to the pictures stored in his phone. Linda, on the other hand, wants to show her pictures to everyone and allows access to her phone as well. Eventually, Mary receives John’s message and goes to the library to join the rest. Kia gives her the password and uses the program running in her laptop to allow Mary to join the meeting. Although there are many other students in the library with BTH devices, most of the time they cannot even discover the team’s devices. Even for the brief periods, during the beginning of the meeting and when Mary joins, that they can discover their devices they cannot connect to any of them and sneak into the team’s meeting.

B. Protocol Design Goals

With this application area in mind we set the following design goals for the protocol:

- Scatternet formation involves preconfigured private piconets with existing security associations and restricts the nodes from connecting arbitrarily.
- Participating nodes should be properly authenticated before associating with the scatternet. New nodes can

security requirements, link-level security should always be complemented by an end-to-end security mechanism.

join the scatternet only by invitation.

- All scatternet traffic is encrypted.
- The scatternet formation should involve minimal (if any) user interaction.
- While the scatternet is connected, the nodes dedicate all of their energy in communicating with each other.
- The protocol should create a topology which simplifies routing.
- Finally, the protocol should be BTH standard compliant and no unrealistic assumptions should be made.

Note that even though we do not restrict the number of nodes, we consider cases with a relatively small number of users (e.g. 10-20), so we do not require scalability to large numbers.

IV. NETWORK FORMATION

A. Scatternet Topology Formation

Our main goal is to design a scatternet formation protocol capable of securely interconnecting private piconets, which may have active sessions among their devices. This requirement prevents the use of a random scatternet formation strategy, because nodes belonging to a private piconet cannot attach arbitrarily in the scatternet. Therefore, our protocol allows only masters of the private piconets, referred to as *Pico-Heads (PH)*², to participate in the scatternet formation; the private piconet slaves do not. At the beginning, a user takes an action and enters the scatternet PIN to initiate the scatternet formation, e.g. like “hosting” a meeting and setting the password. The PH of that user’s piconet becomes the scatternet’s ROOT and initiates the construction of the tree topology by starting scanning³. Note that apart from bootstrapping the process the ROOT has no other special role. Other users wanting to participate also need to take an action, e.g. like “joining” the meeting and entering the password. The PH’s of those users’ piconets start inquiring and look for a scanning node to attach. On successful inquiry the PH pages the discovered scatternet PH, which then authenticates the paging device using the scatternet PIN. If the authentication succeeds, the paging device connects to the scanning node in the scatternet and performs a role switch to become the slave in the newly formed connection and overall a master/slave bridge. Upon attaching to the scatternet each PH starts scanning and becomes a possible attachment point for other free PH’s trying to connect.⁴ The ever increasing number of possible attachment points acceler-

²Note that we allow private piconets that consist of only one node, which is in this case the PH.

³In the rest of the paper we use “scanning” to refer to both inquiry and page scanning.

⁴A PH may decide not to scan if it has reached the maximum number of links it can form. In the rest of the paper we will refer to “nodes scanning”, with the understanding that they may choose not to scan if they have no links available.

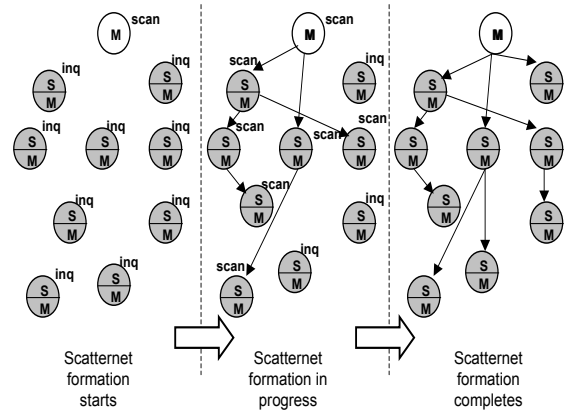


Fig. 1. An example of scatternet formation. Only Pico-Heads are shown.

ates the scatternet formation. This process is incremental and involves a growing *main tree* that contains the ROOT and a number of free PH’s trying to attach to it. Once the scatternet is formed (e.g. as signaled by a timer or a user action), the devices stop scanning and dedicate 100% of their resources in communication. New nodes can then join only by invitation from attached nodes as described in Section IV-B. An example of the scatternet formation process is depicted in Figure 1.

Because every newly connected PH becomes itself a possible attachment point, our protocol creates fully connected scatternets with a high probability, even when not all nodes are within range of each other. Furthermore, it does not put the burden of coordinating the scatternet formation on any particular PH. As far as security is concerned, the protocol always authenticates the PH’s before allowing them to join the scatternet and allows for encryption of the inter-(private)piconet and intra-(private)piconet traffic with separate keys. Data transfer among nodes in the same private piconet is encrypted with a key known only to them and each PH is responsible to allow traffic to flow between the scatternet and any of its private piconet members based on the user’s selection. After the scatternet formation is complete no nodes are scanning, which makes the scatternet undiscoverable and unconnectable, further protecting it from intruders. Although our approach does not address all security threats (e.g. if someone steals the common PIN during scatternet formation, an additional mechanism such as the one described in [10] would be necessary), our protocol considers and takes advantage of the standard link-level BTH security mechanism when forming scatternets.

B. Scatternet Topology Update

Our protocol allows new user arrivals. New nodes can join only by invitation, which means that an already par-

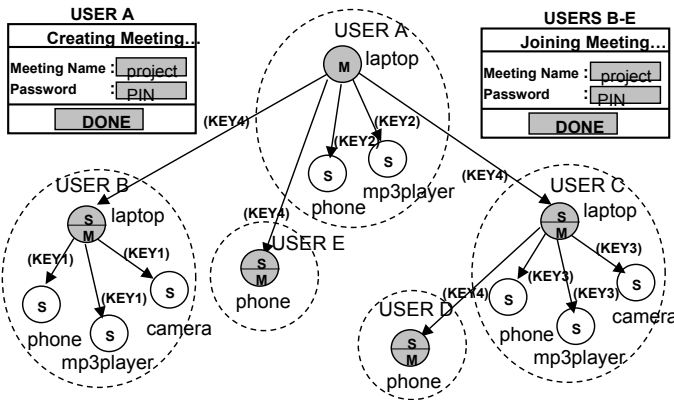


Fig. 2. The user's perspective of the scatternet formation.

icipating user has to take an action, e.g. like “updating” the meeting. That user’s PH broadcasts an UPDATE scatternet message to all PH’s in the scatternet and starts scanning. Upon receiving the UPDATE message each scatternet PH starts scanning too. On the other hand, the new user joining the scatternet has to take an action, e.g like “joining” the meeting and enters the password. That PH starts inquiring and upon discovering a scanning PH it pages it and connects to it after proper authentication using the scatternet PIN. Finally, as in the case of the scatternet formation, the duration of the topology update can be determined based on a timer or some user action.

The downside of allowing scatternet updates is that the scatternet has to briefly become discoverable and connectable. This would open the door to an intruder if the scatternet PIN has been compromised. This topic is part of our future research.

C. The User Perspective

In the protocol description above we mention that the users have to take some actions, which are translated into scatternet protocol events. In this section we summarize the user operations⁵. Figure 2 depicts an example of the user’s perspective of the scatternet formation.

- *Create Meeting*: A user responsible for initiating the meeting performs this operation. She sets the meeting password and performs the “create” operation on one of her devices. This user’s PH becomes the ROOT of the scatternet and starts scanning.

- *Join Meeting*: All users willing to join a meeting perform this operation. Each one has to set the meeting password and perform the “join” operation on one of his/her devices. Each user’s PH starts inquiring and per-

⁵Note that “users” may actually be agents acting on behalf of the real users.

forms the operations described earlier to attach to the scatternet.

- *Update Meeting*: A user in an already formed scatternet performs this operation. Just a simple action, no need to enter the password again. That user’s PH broadcasts the UPDATE message and starts scanning.

D. IP Layer Creation Using PAN profile

Because our protocol builds a loop-free topology, it allows for the creation of an Ethernet local-link on top of a scatternet just by using standard PAN profile [3] functionality, without the need of an additional ad-hoc routing protocol. The PAN Profile supports IP traffic using Ethernet encapsulated as Logical Link Control and Adaptation Protocol (L2CAP) payload based on the BTH Network Encapsulation Protocol (BNEP) [11]. It specifies that a piconet Master will relay packets among its Slaves (and any external LAN connection if present) by using the IEEE 802.1d Ethernet bridging protocol [12]. Nodes supporting the PAN profile also must support IP autoconfiguration [13] to assign themselves IP addresses in ad-hoc scenarios when DHCP services are not present.

V. PERFORMANCE RESULTS

We implemented a prototype of our scatternet formation protocol using BTH v1.1 compliant hardware. We used devices with Cambridge Silicon Radio (CSR) chipsets (Host Controller Interface (HCI) Ver:1.1 (0x1), HCI Rev:0x72, LMP Ver:1.1 (0x1), LMP Subver: 0x72), running Linux kernel 2.4.18 with the BlueZ stack v2.2. We equipped nodes with dual-radios, since no off-the-self BTH hardware supported master/slave (or slave/slave) scatternet operation. This approach, although not optimal, is a useful workaround [14] which represents an upper performance limit. To emulate the effect of having only a single-radio, we always put on hold-mode one radio until the other radio completed its operation.

For our simulation results we modified the Blueware ns-simulator [15], which was in turn based on BlueHoc [16]. We found that even though Blueware and the results in [5] are based on more realistic assumptions than many other results in the literature, some further modifications were necessary. Our changes were based on our experience with real BTH hardware and can be summarized as follows: we implemented a periodic page scan mode with period of 1.28 s and window 11.25 ms, as opposed to nodes entering the page scan state only after responding to an inquiry; we increased the page timeout value from 1.28 s to 6.4 s; we chose randomly between Train A and Train B at the starting point of inquiries; we allowed for fewer than seven BTH links per node to reflect hardware limitations⁶; we randomized the starting

⁶The number of maximum BTH links per node in our results was

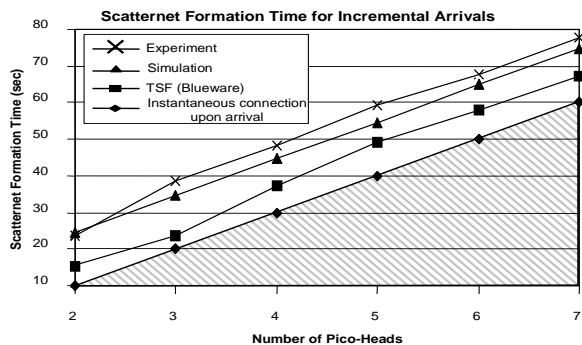


Fig. 3. Scatternet formation time for incremental arrivals.

times of nodes inquiring to avoid artificial synchronizations; we increased the number of inquiry responses parameter from 1 to 10 and allowed for fixed-time inquiries in multiples of 1.28 s. We believe that these changes make our simulation model depict better realistic BTH hardware behavior.

First we examined the scatternet formation delay when piconets arrive incrementally. Choosing incremental arrivals every 10 s allowed us to measure experimental results using our prototype implementation. Each point in Figure 3 shows the overall formation time of a scatternet consisting of n piconets, after the n -th piconet has arrived. In both simulation and experimental results of our protocol, the inquiry length parameter was set to 12.8 s. For comparison we also show the results of TSF [5], as these were obtained running the original Blueware [15] simulation model. When we used the same Blueware model to examine our protocol we got faster formation results than TSF, because in our approach user-actions dictate which nodes will initially perform inquiry and scanning, while in TSF nodes have to go through a period of alternating between inquiry and scanning unassisted by users. However, we believe that the results of our protocol using our modified-Blueware simulation model as shown in Figure 3 are more representative of realistic BTH performance.

Finally, we examined the properties of the resulting scatternet tree topology. In multi-hop networks, the path-length or hop-count between any two communicating nodes influences the end-to-end latency. Figure 4 compares the average path-length of the resulting tree topology of our protocol to TSF. To keep the number of nodes equal, we considered private piconets with only one node (the PH). The results show that our protocol leads to a more balanced tree topology.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we proposed a simple scatternet formation protocol to enable secure ad-hoc group collaboration to four to reflect the used hardware.

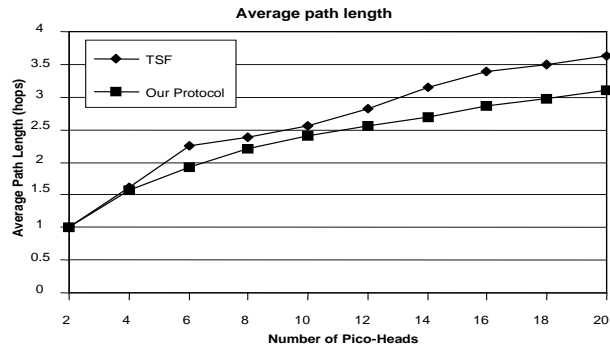


Fig. 4. Average scatternet path length.

Our protocol allows individuals to participate in the scatternet with their private piconets, while maintaining existing sessions and security associations among their devices. Our protocol requires BTH-level authentication before allowing new piconets to join the scatternet and allows separate encryption keys for inter-(private)piconet and intra-(private)piconet links. Once the scatternet is formed the devices dedicate 100% of their resources communicating, making the scatternet undiscoverable and unconnectable in the steady state. We also provided a mechanism to allow new users to join an already formed scatternet after appropriate authentication. Finally, our protocol builds a loop-free, compact tree topology, which allows for simple routing using existing PAN profile functionality, without the use of additional ad-hoc routing protocols. We developed a prototype implementation of this framework and provided initial experimental and simulation results that demonstrate the feasibility of our approach using actual BTH hardware.

Although our approach considers security when forming scatternets, we believe that many issues related to security in BTH scatternets are open. As a future direction, we plan to expand our protocol to address the security threat described in [10], where an intruder discovers the scatternet PIN and tries to join the scatternet and compare our protocol with other secure protocols found in the literature. An additional challenge is dealing with dynamic environments, when users can arbitrarily come and go at any time causing partitions in the scatternet and requiring a protocol for healing, which makes static solutions inapplicable. Finally, we are planning to address the issue of selectively and dynamically allowing access to specific scatternet devices.

REFERENCES

- [1] Bluetooth Special Interest Group. *Specification of the Bluetooth System, v1.1*. February 2001.
- [2] Cylink Corporation. *SAFER+ (Secure And Fast Encryption Routine) Encryption Algorithm*. <http://www.cylink.com/library2/downloadbody.htm>.
- [3] Bluetooth Special Interest Group. *Personal Area Networking Profile, v1.0*. July 2002.

- [4] G. Zaruba, S. Basagni, and I. Chlamtac. Bluetrees - scatternet formation to enable Bluetooth-based ad hoc networks. In *IEEE Int. Conf. on Comm. (ICC'01)*, 2001.
- [5] G. Tan, A. Miu, J. Guttag, and H. Balakrishnan. An efficient scatternet formation algorithm for dynamic environments. In *IASTED Comm. and Comp. Networks (CCN'02)*, 2002.
- [6] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaira. Distributed topology construction of bluetooth personal area networks. In *IEEE INFOCOM*, 2001.
- [7] C. Petrioli and S. Basagni. Degree-constraint multihop scatternet formation for Bluetooth networks. In *IEEE Globecom*, 2002.
- [8] C. C. Foo and K. C. Chua. Bluerings - bluetooth scatternets with ring structures. In *IASTED International Conference on Wireless and Optical Communication (WOC'02)*, 2002.
- [9] F. Cuomo, G. Di Bacco, and T. Melodia. SHAPER: a self-healing algorithm producing multi-hop Bluetooth scatternets. In *IEEE Globecom*, 2003.
- [10] Karl E Persson and D. Manivannan. Secure connections in bluetooth scatternets. In *In Proceedings of the 36th Hawaii International Conference on System Science*, 2003.
- [11] Bluetooth Special Interest Group. *Bluetooth Network Encapsulation Protocol*. February 2001.
- [12] IEEE Std. *ISO/IEC 10038:1998 [ANSI/IEEE Std 802.1D] Information technology-Telecommunications and information exchange between systems-Local area networks -Media Access Control(MAC) bridge*. 1998.
- [13] S. Cheshire and B. Aboba. *Dynamic Configuration of IPv4 Link-local Addresses*. Internet draft Zeroconf, March 2001.
- [14] M. Leopold, M. Dydenborg, and P. Bonnet. Bluetooth and sensor networks: A reality check. In *ACM SenSys*, 2003.
- [15] G. Tan. Blueware: Bluetooth simulation for ns. <http://nms.lcs.mit.edu>.
- [16] IBM Research. Bluetooth extension for ns, February 2001. <http://www-124.ibm.com/developerworks/opensource/bluetooth/>.