SecureWear: A Framework for Securing Mobile Social Networks

Baishakhi Ray and Richard Han

University of Colorado, Boulder,
Department of Computer Science
baishakhi.ray@gmail.com, rhan@cs.colorado.edu
http://www.cs.colorado.edu/

Abstract. With the rising popularity of social networks, people have started accessing social networking sites from anywhere, any time, and from a variety of devices. Exploiting this ubiquity, the social networking data of an individual can be coupled with her geographical location to build different context aware applications. However, the existing infrastructure to share this personalized data forces users to compromise their privacy. In this paper, we present a secure framework, which allows interaction of social network information with location-based services, without compromising user privacy and security. Through exchanging an encrypted nonce ID (EID) associated with a verified user location, our framework allows location-based services to query its vicinity for relevant information without disclosing user identity. We further argue that, this kind of framework should be adopted as a common security framework for mobile-social interaction to meet privacy requirements.

Keywords: mobile computing, social network, security, privacy, wearable device.

1 Introduction

With the advent of mobile social networks [12,6], an exciting new paradigm emerges, in which, local environments with numerous electronic and mobile/wireless devices are bathed with social networking information. Context-related information and services are increasingly used for spontaneous socializing and collaboration. However, when users participate in such communication, they unconsciously leave private information traces that can undermine their privacy.

In this work, we explore one such simple application that notifies the user whenever her Facebook friend is nearby. The application is deployed on smart phones as well as on a electronic shirt. When running on a cellphone, it notifies the user, which friends are located nearby. In case of wearable shirt, the application indicates if someone is around by lighting an embedded LED (green for a friend, red for foe).

This application represent a larger class of context-aware applications in which both the current location of the user as well as their social networking information are extracted to provide context. In this way, the user leaves trace of his identity which can be misused. In particular, mobile social networks introduce security and privacy challenges. A user may not always want to reveal their identity, location, or preference

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 515–524, 2012.

information. Moreover, a user may wish to reveal different part of their preference depending on their location, time, mood, presence/status of other users, etc. Wearable MoSoNets are even more challenging from a security and privacy point of view due to extreme resource and user interface constraints associated with embedded wearable items.

The contribution of this work is to introduce a common security framework for resource-challenged mobile social networks. We demonstrate an implementation of the framework on smart-phones and wearable shirts. The framework provides a general architecture that can be extended to any mobile device capable of communicating over wireless media.

2 System Architecture

Let us first introduce a sample application area to motivate the assumed system architecture. Consider Maya is an undergraduate student who wants to find a classmate from her algorithm class to share some thoughts on a homework problem . She can set an alarm on her mobile device, where if a classmate is around, she will be notified. Now, let's assume the algorithm class buddies have a Facebook group for their class discussion. Hence, Maya's mobile alarm application has to retrieve the Facebook IDs of all students registered in the group and search locally to find a match. Such an application would need a system that could announce wirelessly the local presence of each user, and link that user identity with their social network profile in order to facilitate communication among matched users.

The basic system thus consists of three components: mobile devices/wearables, an access point(optional), and a social network. Our particular implementation uses electronic shirts as the wearables, a laptop as the local access point, and Facebook as the social network. We chose wearables over mobile devices to show that our framework is suitable for resource constrained devices as well.

The shirts and laptop communicate with each other wirelessly via Bluetooth and the laptop communicates with Facebook via an internet connection. In the case of smartphones, we can omit the requirements of access points, as the smartphones can use their own wifi or cellular services. The overall system model is shown in Figure 1.

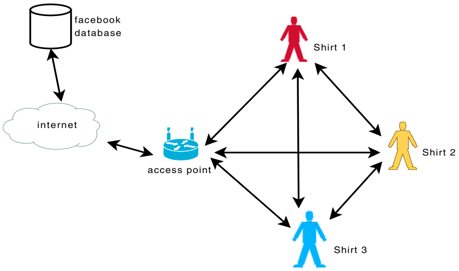


Fig. 1. Basic System Model

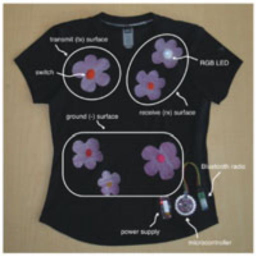


Fig. 2. Wearable Shirt

2.1 System Model

The eShirts. Each wearable shirt contains an embedded micro-controller, a Bluetooth radio, and an RGB-LED. Commercially available LilyPad Arduino modules [8] were stitched to each shirt using conductive threads. Such a shirt is shown in Figure 2. Each electronic shirt (e-shirt) broadcasts its Facebook ID periodically using the embedded Bluetooth device. Bluetooth was chosen due to its limited range and low power usage.

The Access point. The access point relays data between the e-shirt and Facebook.com. It must be able to send and receive data to and from the e-shirts and run code that accesses the Facebook APIs [1].

We have implemented the access point functionality on a laptop as well as on phones (Nokia N80). Cellphone-based access points allow users to move from place to place, without disrupting ongoing communications. in addition to e-shirts, cellphones can also serve as the end client communication devices. We have used the Java 2ME Wireless Toolkit to program the cellphone and J2SE for the laptop.

Social Network Information. We have selected the Facebook database as our backend data server. We retrieved social networking information through java library of Facebook API [1].

In this application, one e-shirt listens for Facebook IDs announced wirelessly by nearby e-shirts and checks Facebook's database to see whether this user is on its friend list. If yes, the embedded LED glows green, else red.

2.2 Threat Model

The basic system architecture described in Section 2.1 models a typical context-sensitive mobile social networking infrastructure. However, existing implementations of such systems compromise users' privacy. For example, a person wearing e-shirt can easily be tracked by listening to its broadcast ID. More over, a malicious user A, can spoof B's ID to falsely register B's presence.

We have identified some of the major threats that existing mobile social systems impose. These are explained by a running example as presented in Figure 1.

1. **Spoofing:** e-shirt1 can spoof e-shirt2's Facebook ID. In that case, if e-shirt1 disguises e-shirt2's ID to e-shirt3, then e-shirt2 will falsely appear to be in e-shirt3's vicinity.
2. **Eavesdropping:** e-shirt2 can steal e-shirt1's password. In the above simplified model, to retrieve data from Facebook, at least once in every session, shirts have to login to the Facebook. During login they have to send Facebook login ID and password information to the access point. If this is not protected by any cryptographic method, anybody eavesdropping on wireless communications by other shirts can steal their login ID and password.
3. **Replay :** Say e-shirt3 eavesdrops on e-shirt1's Facebook ID. As e-shirt1 is broadcasting its ID, this will reach e-shirt3 and e-shirt3 can simply forward the packet to any remote location to make it appear as if e-shirt1 is in that location.
4. **Compromised shirt:** An e-shirt itself can be compromised to the extent that the advertised embedded Facebook ID is a fake one.

5. **Energy exhaustion:** e-shirt1 can continuously broadcast its own Facebook ID. Listening that every time e-shirt2 will query the Facebook database through access point, thus losing considerable amounts of energy. Alternatively, e-shirt1 can continuously spoofing several other Facebook ID and send them to e-shirt2. Thus e-shirt2 will query Facebook to resolve each received Facebook ID every time. Thus all its energy will be consumed and ultimately it will shut itself down.

To handle the above discussed threats, we propose a security framework in section 3.

3 Security Architecture and Solutions

As identified in Section 2.2, security and privacy vulnerabilities are great challenges for context-sensitive mobile social networks. To provide security, the simplest approach would be to use some standard cryptographic techniques, where each wireless mobile social device, say an e-shirt or cellphone, will encrypt its Facebook identity before broadcasting it. The other nearby devices, if legitimate users, will decrypt the received packet and retrieve the ID. The major problems with this approach are:

1. How do you know who is a legitimate user? If a person buy an e-shirt, registers its Facebook ID through some means and starts using it, how will it become a valid user? Here the concept of a trusted third party authentication comes into the picture, which can endorse the validity of an electronic shirt.
2. Secondly, running all the standard cryptographic algorithms on embedded systems like e-shirts requires a significant amount of processing power and energy consumption. As the computational power of the wearable chip is quite constrained, and it runs on a small battery, this limits the use of standard security algorithms.
3. Broadcasting authenticated data to all the wearable shirts in the vicinity is even more challenging. Established authentication policies rely on either asymmetric digital signatures method or on purely symmetric solutions [20]. The asymmetric method is impractical in wearable domain; one of the main reasons are that long signatures with high communication overhead requires 50-1000 bytes per packet, very high overhead to create and verify the signature. The symmetric method is also infeasible: Gennaro and Rohatgis initial work required over 1 Kbyte of authentication information per packet [14], and Rohatgis improved k-time signature scheme requires over 300 bytes per packet [22].

In this paper we have proposed a new security framework, particularly suitable for the mobile social network domain. We have introduced a proxy server **eShirt.com**, whose role is to authenticate each of the local wireless devices. In our example, eShirt.com will validate each e-shirt. If e-shirts pass the validity check, then eShirt.com contacts Facebook to retrieve the requested information and sends that back to the concerned e-shirts.

In Section 3.1 the new security model is discussed and 3.2 analyses how this model takes care of all the threats identified previously.

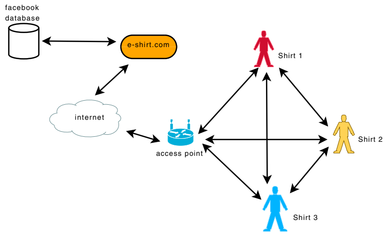


Fig. 3. System Model introducing eshirt

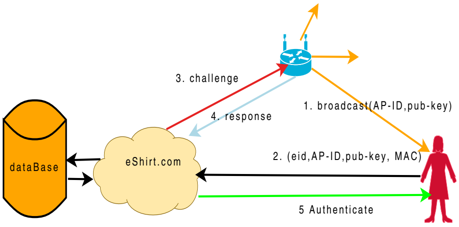


Fig. 4. AP Authentication Message Flow

3.1 Security Model

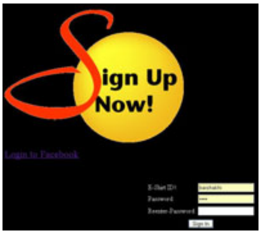
As shown in Figure 3, we have introduced an authentication server **eShirt.com** in between the access point and the social-network database, which is Facebook in this case. The whole communication can be categorized into six phases:

1. Registering an e-shirt to eShirt.com
2. Registering Access point to eShirt.com
3. Communication between local access point and eShirt.com
4. Communication between two e-shirts.
5. Communication between an e-shirt and local access point.
6. Communication between eShirt.com and Facebook.

Registering to eShirt.com. The user has to register and activate a new e-shirt, to the trusted 3rd party proxy server eShirt.com, before he starts using it. The registration process requires the user to login to Facebook, so that their Facebook-ID can be retrieved and it needs the username, password to register to eShirt.com. See Figure 5. We believe that users wishing to exploit the unique power of mobile social networks will be willing to divulge this information to the mobile social network provider. For example, kids desiring the fun of being able to use the e-shirt to identify friend/foe would be willing to let an eShirt provider know who are their friends on Facebook. This is of course a simplified example, and we imagine much more sophisticated applications that exploit the power of mobile + social interactions. Once registered, each user will get a unique eShirt.com ID. From now on, we will term this eShirt.com ID as EID. corresponding to each EID, an unique MAC-Key is generated using CBC-MAC[20]. The user has to download both EID and MAC-Key in order to activate the e-shirt and it will be embedded in the shirt unless reloaded again. The user can login to eShirt.com as shown in Figure 6 and retrieve all its relevant information later.

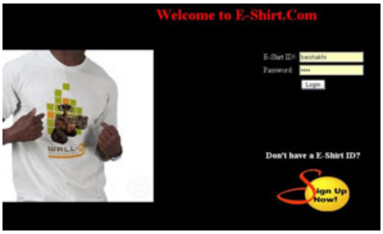
Corresponding to each user, the eShirt.com will maintain a database which is shown in table 1.

Access point authentication: Each AP has to register itself with eShirt.com and eShirt.com will give it an unique identifier (AP-ID), a digitally signed public key and a private key as shown in Table 2.



E-Shirt ID: baishakhi
Password: ***
Reenter-Password:

Fig. 5. Registration Page to eShirt.com



E-Shirt ID: baishakhi
Password: ****

Fig. 6. Login Page to eShirt.com

Table 1. eShirt.com database for e-shirt

user	password	Facebook-ID	eid	MAC-key

Table 2. eShirt.com database for AP

login	Password	AP-ID	Pub-key	Priv-key

When an e-shirt wants to establish a connection with the AP, the e-shirt will select a random AP based on the availability. The AP broadcasts a signed public key along with its AP-ID to the e-shirt. The e-shirt will tag its own EID with the received message and do a MAC operation to received message i.e,public key and AP-ID and EID, with the stored MAC-key and send it to eShirt.com along with EID. The table 3 and 4 show the corresponding packet formats.

Table 3. AP Authentication Packet

EID	AP-ID	Pub-key	MAC

Table 4. packet from AP to e-shirt

AP-ID	signed-public-key

When eShirt.com gets the digitally signed public key and AP-ID, it verifies it with the received MAC message to avoid tampering. It will then validate the (AP-ID: public-key) pair against its own AP database 2. If matches, it sends a random secret to the AP by sending a message encrypted by the public key of the AP. The AP should decrypt the message with its private key and send it to the eShirt.com encrypting through its private key. If the eShirt.com can decipher it, it'll send a signal to the related e-shirt (encrypted message) authenticating the AP. Figure 4 shows the entire communication that takes place to select an authenticated AP.

e-shirt to e-shirt talk. Once the e-shirt selects an authenticated AP, it can start to communicate with its neighboring e-shirts. Our security model ensures the following basic two security features:

- Even if the attacker can see the packet, it cannot derive any useful information.
- It has to be endorsed that the packet is coming from the proper user, so nobody can even eavesdrop on the packet.

An e-shirt will synchronize its time and location with the selected AP. An e-shirt will then keep a track of time through some internal counter increment and synchronize itself with the AP periodically. If the e-shirt changes its location, it has to select another

AP through the same authentication procedure and has to be synchronized with the new AP. In the new security scheme, instead of broadcasting Facebook-ID, each eshirt broadcasts its EID along with a unique monotonically increasing number called nonce, time-stamp and location data (as available from AP). These four fields constitute our basic data communication unit.

$$D_{org} = (EID, NONCE, TIMESTAMP, LOCATION); \quad (1)$$

$$E = D_{(K_{encr}, nonce)}, \ where \ D = (nonce, timestamp, location); \quad (2)$$

$$M = MAC(K_{mac}, nonce|D) \quad (3)$$

The last three fields are encrypted as they can be retrieved easily by the eShirt.com if the EID is known. To maintain data integrity, a MAC operation will be performed on the whole data, i.e., EID and the encrypted data, by the MAC key stored in the e-shirt. To attain the security constrains as discussed above, we propose a packet structure shown in Figure 7. The K_{mac} and K_{encr} are derived from the original key embedded in the e-shirt, as per the method described in Section 4.

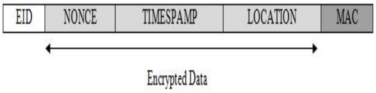
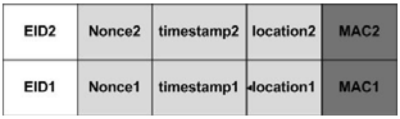


Fig. 7. eShirt-eShirt talk : Packet Structure



EID2	Nonce2	timestamp2	location2	MAC2
EID1	Nonce1	timestamp1	location1	MAC1

Fig. 8. packet received at eshirt.com

Thus the complete message that is sent from e-shirt1 to e-shirt2 is as follows:

$$Shirt1- > Shirt2 : D, MAC(K_{mac}, nonce|D) \quad (4)$$

verification at e-shirt.com. Once an e-shirt, say e-shirt2 receives the broadcast data from e-shirt1, it tags its own packet of same format as 2 and sends it to the eShirt.com through local access point. The eShirt.com receives the packet shown in figure 8 from e-shirt2.

In Figure 8 , 1 corresponds to e-shirt1 (the broadcaster), and the 2 corresponds to the receiver e-shirt2. Once eShirt.com receives this packet, it'll first check the credentials of e-shirt1, i.e., EID1. eShirt.com knows the K_{mac} corresponding to EID1, and thus it can check the received MAC. If correct, then it knows that the content contained within the packet is from e-shirt1 and has not been tampered with. It then checks the fields corresponding to e-shirt2, and if incorrect discards the packet as not coming from e-shirt2. If the credentials of e-shirt2 also match, eShirt.com now knows that the packet came from e-shirt2 and has not been tampered with. It then checks from the Facebook database whether these two shirt owners are friends or foes. If friends, eShirt.com sends a secure encrypted message to e-shirt2 to glow its green light, else glow its red light.

Along with identity verification, eShirt.com checks the corresponding time-stamp and locations between the two shirts. If the timestamps do not match within certain error limit, then eShirt.com discards the packet. This is based on the assumption that e-shirt2 can receive a pretty old packet from e-shirt1 and by then e-shirt can move out of the environment. Similarly, if the geographic location doesn't match within certain error limit, eShirt.com simply discards the packet.

3.2 Threat Prevention

In this section we will discuss how our proposed security model takes care of the threats discussed in section 2.2.

1. **Spoofing:** As the Facebook ID is not embedded in any shirt, the Facebook ID cannot be spoofed. Even if some attacker spoofs EID (eshirt id), that will not help as every time the credential is checked by eShirt.com against the MAC key and unique nonce. As the MAC key is secret and is known only by the eShirt.com server and corresponding e-shirt, no outsider can retrieve that. So spoofing is not possible.
2. **Eavesdropping:** This is taken care by the fact that there is no login to Facebook in every session. In fact in this scheme the direct interaction between e-shirt and Facebook is not required anymore. As mentioned, only in the registration process a user has to give his/her Facebook ID to to the eShirt.com server.
3. **Replay:** If e-shirt3 is an eavesdropper and it's forwarding e-shirt1's packet to e-shirt2, which is in different environment, then the eShirt.com can detect the theft. First the server will check the time-stamp value of both and then the location. If e-shirt1 and e-shirt2 are in different location, then this will be caught. If the eavesdropper forwards an old packet, that can also be caught from the time-stamp value. In the case of a replay attack, eShirt.com will inform the two shirts in concern. Then they can backoff their transmission for a certain time.
4. **Compromised shirt:** If the shirt itself is compromised and the user can suspect that, it can nullify its eShirt.com ID and re-register again.
5. **Energy exhaustion:** As discussed, if a shirt is attacked by some malicious user, eShirt.com can detect that and it'll let the shirt know about the attack. The shirt can switch itself off for a certain time and then again check its environment through normal operation. This can help mitigate an energy exhaustion attack.

4 Implementation and Results

One of the major issues we need to keep in mind while implementing our security protocol in eshirt is the limited computing power and memory in the eshirt. To save memory we use a single block cipher for encryption, decryption and MAC computation as suggested by [20]. We have selected RC5 as the block cipher due to low computational and memory requirement of the cipher [20]. We use our own implementation of RC5 algorithm instead of using the standard openssl one as the overheads of the openssl implementation is too large for our purposes. For encryption and decryption, we use RC5 in CTR mode and for computing MAC we use CBC-MAC as suggested by Perrig et al. [20]. In our experiment we found, the average round-about time for detecting a new e-shirt and glowing the LED based on friend and foe information is around 1.5 sec.

5 Background and Related Work

Social Networks. According to Facebook.com's statistics page, the site has over 750 million active users [2], making it the 5^{th} most trafficked website in the world. Existing social networks already allow users to share a rich set of contextual data online. The amount of data and the type of data stored on these sites is growing daily. Remarkably, Facebook alone gets over 14 million new photos uploaded to it every day. Also, the way in which user access that data is expanding. Facebook applications allow developers to provide users with new ways of accessing data. More than 95% of Facebook users have at least one Facebook application installed [2]. The existence of such data and access to it through the Facebook API allow for a simple touch sensor and LED to become a gateway to an entire world of applications and information. In effect, the wearable and its use become a part of a users digital identity.

Wearable Sensing and Actuation. A large body of work has explored wearable sensing platforms that detect information about a wearer's movements, physiological state and location, primarily to support context aware applications [13,18,19]. Another class of projects have investigated how communicating wearable devices can augment real-world interactions and experiences. For example, Borovoy et al. developed communicating name tags that tracked wearer's face to face interactions and allowed wearers to easily exchange virtual business cards [5]. A final class of relevant devices has employed wearable and textile-based actuators for a variety of purposes. Several designers have used LEDs in wearables displays [9,3,11,4]. Our system differs from previous ones in its integration of wearable, socially relevant sensing and actuation with a social networking system.

MoSoNets & TechoSharking. Some existing work in "MoSoNets" has merged social networks with mobile devices. Dating application on mobile phones [17] and a Bluetooth-based presence sharing system called Serendipity [12] have combined local and social information, but the outcomes are fairly closed systems which are cut off from the wider online social networking phenomenon. Research in context-aware smart spaces [15] and context-aware UIs [16,10,21] hint at some of the possibilities of MoSoNets, but are largely disconnected from the phenomenon of online social networks. However this approach does not manage to exploit the richness of information existing both in the social network and integrate this with the physical actions of the user.

However, our previous work [6,7] tried to bridge this gap by accessing social networking information from smart phone and wearable devices. But they have not investigated the security and privacy threats inherent in such application.

6 Conclusion

Research on the intersection of mobile computing and social networks is still rather immature. The integration of wearables into a MoSoNet demonstrates the extent to which these new technologies may be integrated in users' physical interactions. While MoSoNets offer the promise of bringing people closer together, they also pose a threat to security and privacy by bringing malicious users closer to people.

The MoSoNet application presented in this paper shows a sample scenario to demonstrate potential privacy threats in cyber-physical interaction and builds a proof-of-concept framework that mitigates such threats.

References

1. Facebook developer resource, `http://developers.facebook.com/resources.php`
2. Facebook statistics, `http://www.facebook.com/press/info.php?statistics`
3. Nyx illuminated clothing, `http://www.nyxit.com/`
4. Berzowska, J., Coelho, M.: Kukkia and vilkas: Kinetic electronic garments. In: Proceedings of the IEEE International Symposium on Wearable Computers (2005)
5. Borovoy, R., McDonald, M., Martin, F., Resnick, M.: Things that blink: computationally augmented name tags. IBM Systems Journal 35(3-4) (1996)
6. Beach, A., Gartrell, M., Akkala, S., Elston, J., Kelley, J., Nishimoto, K., Ray, B., Razgulin, S., Sundaresan, K., Surendar, B., Terada, M., Han, R.: Whozthat? evolving an ecosystem for context-aware mobile social networks. IEEE Network 22(4), 50–55 (2008)
7. Beach, A., Ray, B., Buechley, L.: Touch Me wE@r: Getting Physical with Social Networks. In: IEEE International Conference on Computational Science and Engineering, vol. 4, pp. 960–965 (2009)
8. Buechley, L., Eisenberg, M., Catchen, J., Crockett, A.: The lilypad arduino: Using computational textile to investigate engagement, aesthetics and diversity in computer science education. In: CHI, pp. 423–432 (2008)
9. Buechley, L., Eisenberg, M.: Fabric pcbs, electronic sequins, and socket buttons: Techniques for e-textile craft. Personal and Ubquitous Computing (2007)
10. Cheverst, K., Davies, N., Mitchell, K., Friday, A.: Experiences of developing and deploying a context-aware tourist guide: The guide project. In: ACM MobiCom, pp. 20–31 (2000)
11. Dunne, L.E., Toney, A., Ashdown, S., Thomas, B.: Subtle garment integration of technology: A case study of the business suit. In: IFAWC (2004)
12. Eagle, N., Pentland, A.: Social serendipity: Mobilizing social software. IEEE Pervasive Computing 4(2) (April-June 2005)
13. Choudhury, T., et al.: The mobile sensing platform: An embedded activity recognition system. IEEE Pervasive Computing 7(2), 32–41 (2008)
14. Gennaro, R., Rohatgi, P.: How to sign digital streams, pp. 180–197. Springer, Heidelberg (1997)
15. Microsoft Vision Group; Microsoft easyliving, `http://research.microsoft.com/easyliving/`
16. Harter, A., Hopper, T., Steggles, P., Ward, A., Webster, P.: The anatomy of a context-aware application. In: Mobicom 1999, pp. 59–68 (August 1999)
17. Iwatani, Y.: Love: Japanese style. In: WIRED (1998)
18. Madan, A., Pentland, A.: Vibefones: Socially aware mobile phones. In: ISWC, pp. 109–112 (2006)
19. Pentland, A.: Socially aware computation and communication. IEEE Computer 38(3), 33–40 (2005)
20. Perrig, A., Szewczyk, R., Wen, V., Culler, D., Tygar, J.D.: Spins: Security protocols for sensor networks. Wireless Networks, 189–199 (2001)
21. Priyantha, N., Miu, A., Balakrishnan, H., Teller, S.: The cricket compass for context-aware mobile applications. In: ACM MobiCom, pp. 1–14 (2001)
22. Rohatgi, P.: A compact and fast hybrid signature scheme for multicast packet authentiation. In: CCS 1999, pp. 93–100. ACM, New York (1999)