Engineering and Scientific International Journal (ESIJ)
Volume 5, Issue 1, January – March 2018
ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

Mobile-to-Mobile Communication using Secured Shared Key

G.S. Gopakumar [#1], R. Anitha [*2]

[1]*Scholar, Master of Computer Applications, S.A. Engineering College, Chennai-77.*
gopakumar0608@gmail.com
[2]*Assistant Professor, Department of Computer Applications, S.A. Engineering College, Chennai-77*

Abstract — In mobile networking, Mobile-to-Mobile communication helps in data exchange between actual and closest devices. The Mobile-to-Mobile communications are used to make short range communication [1]. It improves the network performance and supports proximity based services. Our primary goal is to provide potential solution to mobile-to-mobile communications. In this paper we use an efficient key agreement protocol known as Diffie-Hellman key and commitment scheme. The proposed protocol has less computation and communication cost, while compared to other protocols.

Keywords — *Diffie-Hellman Key; Wi-Fi Direct Protocol; MANA Protocol.*

1. Introduction

Huge amount of data traffic is generated by personal devices such as tablets and smart phone mobiles by using internet and downloading different applications. Establishment of Mobile-to-Mobile communication is used to unload traffic burden in personal devices [2]. To provide a wireless communication, mobile users use Device to Device (D2D) technology. Here, we use Mobile-to-Mobile links between cellular users to upgrade the network performance while transmission [3].

The Wi-Fi Direct which also known as Wi-Fi Peer to Peer (P2P) is a Wi-Fi standard which enables Mobile-to-Mobile connections with Wi-Fi frequency band. Wireless channels are known to be vulnerable to different kinds of attacks, so the security is a major factor for the Mobile- to-Mobile communications [4]. The first step is to introduce a shared secret key to make safe communication between two mobile users during Mobile-to-Mobile link connection. Without any prior knowledge the Diffie-Hellman key protocol enables mobile devices together to setup a shared secret key. For example, here we introduce two devices A and B. To mutually authenticate the devices A and B, they interchange their public keys over a secure channel. To mutually authenticate this protocol, we need a huge number of bits.

The existing protocol such as MANual Authentication (MANA) protocol [5] decrease the size of the authentication message to k bits, but it requires stronger authentication channel. To overcome this, we introduce a commitment scheme based on 3-round key agreement protocol [6]. The proposed protocol has less computation and communication expenses compared to the existing protocol, and also pursuing the same security level. In this paper we survey the security issues and challenges in mobile-to-mobile communication and we propose efficient Diffie –Hellman key agreement protocol in a Wi-Fi direct protocol and execute it on android mobiles.

2. Existing System

In the existing system, Public Key Structure (PKI) is a standard approach which is generally used [7]. To authenticate messages, they use private and public keys. In traffic analysis, cryptographic methods are vulnerable to protect messages. We can achieve anonymity, privacy, confidentiality unlink ability and integrity while exchanging messages between mobile users. For example, by measuring the transmission rate with detection of source and destination, the message path can be easily unveiled.

2.1 Disadvantages of Existing System

- Computer system or network can be affected by an intruder even though it is powerfully encoded.
- One of the major aspects of data security is high availability, which cannot be ensured by the use of cryptography.
- Other techniques are used to defend against attacks such as Denial of Service (DOS) attack.
- Cryptography does not defend against vulnerabilities which occur from low design of method, protocols and infrastructures.
- Cryptography is more costly to implement.
- To use Public Key cryptography and to maintain the public key infrastructure are costly.

3. Proposed System

The Diffie-Hellman key agreement protocol is used to establish a shared secret key between two android mobile users. This protocol is used to communicate through an insecure channel. Along with this, we use a commitment scheme, i.e., the recipient can reveal the committed value after the sender explored to it (i.e., commitment scheme is used to hide).

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

3.1 Advantages

- Diffie-Hellman key interchange algorithm agrees two users to start communication over a vulnerable transmission channel.
- Diffie-Hellman key has less communication and computation cost.
- It is more secure than the other cryptographic mechanisms.

4. Methodology

For Mobile-to-Mobile communication, two mobile users should establish a shared secret key. Both of them use smart phone which is adept for connecting through a wireless channel. To perform a Diffie-Hellman key agreement protocol, both the devices should have a computation capacity and should be able to display sequence of digits. The two users do not have any pre-shared cryptographic knowledge and infrastructure. To authenticate, the message can be visually or verbally recognized by the divices.

It permits one user to commit to a selected value while keep it invisible from others, with ability to unveil the committed value later. The user cannot alter the value after committed to it. The commitment scheme is used to hide i.e. the recipient can only realize the committed value after the sender 'opens' it. It is explained by two algorithms 'Commit' and 'open':

In this consequence consider *x* as commit, *y* as decommit and *z* as message. Commit: *(x, y)*←*z* transforms a value *z* into a commit/open pair *(x,y)*.The commit value *x* reveals no data of *z*, but with a decommit value *y* together *(x,y)*will unveil *z*. Open: *z*←*(x,y)* output original value *z* if *(x,y)* is the commit/open pair generated by Commit(z). It is visually represented in figure 1.

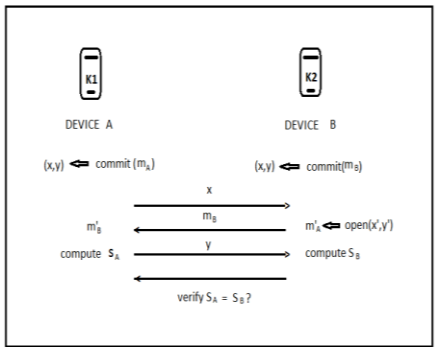

Fig.1: Secure key exchange protocol

5. Working Model

First, we have to insert the smart card in the card reader terminal, within a second it will recognize our overall details by matching the details in the database by using Secure Socket Layer (SSL) or Hyper Text Transfer Protocol (HTTP) network and it takes the Internet protocol (IP) address from the public server and processes it to display in the monitor. With the help of the chip presented inside the silver coated part in the smart cards, it does all these operations. It will display the username or ID and ask for the authentication like passwords or pins. Those passwords will also be checked with the database for the specific user. If the data provided is correct, it will proceed for the further steps or it will be terminated when the password is wrong. Once the password we entered is correct, then it asks for what type of transactions we needed. Then it will be done and the system will be terminated or the connection will be closed once the process is completed.[4]

6. Implementation

In this paper, we introduce a key agreement protocol in Wi-Fi Direct protocol. We call this as a "Secured Wi-Fi Direct protocol", which provides a secure shared key setup between two android mobiles. The execution output shows that the two mobiles obtained same secured shared key in Mobile-to-Mobile communication. By using Wi-Fi frequency, without the help of any access points, the Wi-Fi Direct protocol permits two parties to setup a Mobile-to-Mobile connection. Figure 2 shows the establishment of Mobile-to-Mobile connection using Wi-Fi Direct. First two devices perform the channel probing and recognize each other. Then two users will perform a three way handshake [3] to discover the group owner for the Mobile-to-Mobile connection. Later the mobile devices have approved on their corresponding roles, to setup the Internet Protocol (IP) address for both devices and a DHCP interchange will be take placed. Thus the Mobile-to-Mobile communication is established. We enhance planned protocol on top of the existing Wi-Fi direct protocol. After the address configuring phase, two devices are considered in proposed key agreement protocol and the mutual authentication process to admit shared secret key will be completed.

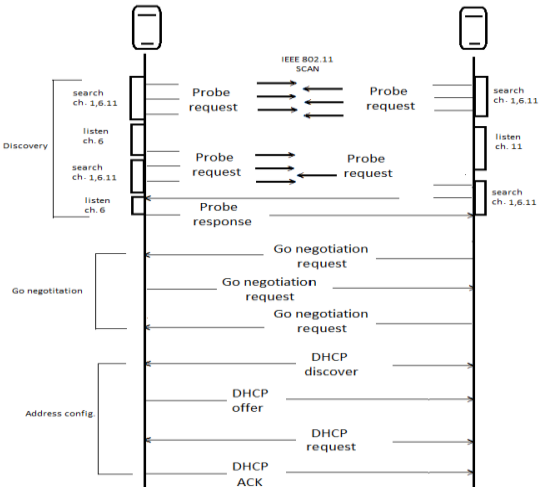

Fig.2: Wi-Fi Direct Protocol

ISSN 2394-7187(Online)
ISSN 2394 - 7179 (Print)

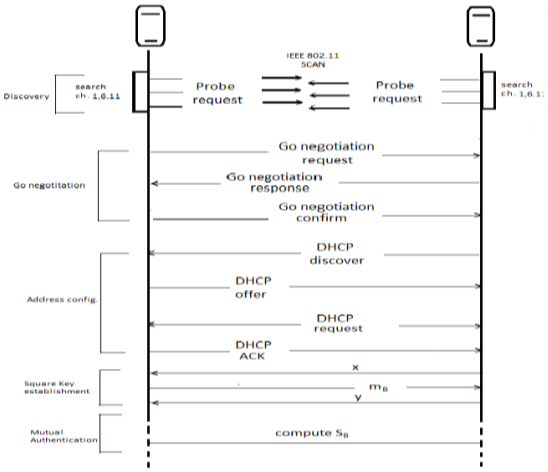

Fig.3: Secure Wi-Fi Direct Protocol

As lengthy as two devices had established on the verification message, they can consequently use their shared secret key for upcoming communication. The device and operating system of the smart phone we used is Samsung Galaxy J1 mobile with the Android KitKat version 4.4 as shown in figure 4. The secure protocol is implemented by programming the Android-TCP-socket.

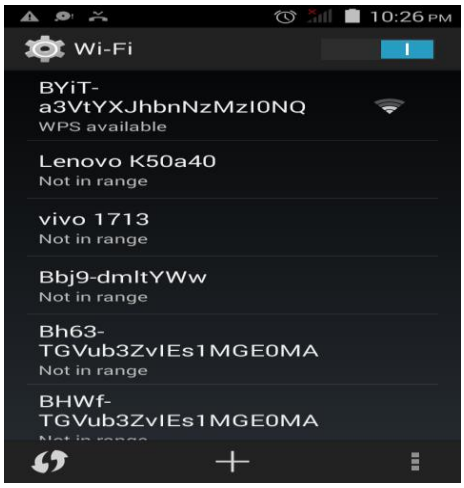

Fig.4: Android Wi-Fi

7. Result

The result is shown in figure 5. We listed some parameters of Diffie-Hellman, as well as hexadecimal authentication N strings. A 40 digits *p* value is used, that gives a roughly 130 bits secret key. The length of the authentication string is set to be 16 bits (4 hexadecimal digits), which is effortless to be correlated by the two participants and to achieve a strong security level. The authentication string is set to be 16 bits which is easy to compare by the two mobile users.

Fig.5: Result

8. Conclusion

In this paper, we have presented a comprehensive overview of results related to Mobile-to-Mobile communication in the emerging latest smart phones. Wi-Fi-direct protocol can be implemented in new upcoming android mobile phones for a secure file sharing. In spite of impressive benefits, Mobile-to-Mobile communication encounter many security threats. Based on the security architecture and security requirements, we reviewed the existing work in order to explore open research issues and propose future research directions.

References

[1] C. Yu, O. Tirkkonen, K. Doppler, and C. Ribeiro, "Power optimization of device-to-device communication underlaying cellular communication," in Proc. IEEE ICC, pp. 1-5, 2009.

[2] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklos, and Z. Turanyi, "Design aspects of network assisted device-to-device communications," IEEE Communications Magazine, vol. 50, no. 3, pp.170-177, 2012.

[3] D.-Q. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, G. Feng, and S.-Q. Li, "Device-to-device communications underlaying cellular networks," IEEE Trans. Commun., vol. 61, no. 8, pp. 3541–3551, Aug. 2013.

[4] M. J. Yang, S. Y. Lim, H. J. Park, and N. H. Park, "Solving the data overload: Device-to-device bearer control architecture for cellular data offloading," IEEE Veh. Technol. Mag., vol. 8, no. 1, pp. 31–39, Mar. 2013.

[5] C. Gehrmann, C.J. Mitchell, and K. Nyberg, "Manual authentication for wireless devices," RSA Cryptobytes, vol. 7, No. 1, pp. 29-37, 2004.

[6] D. Zhu, A.L. Swindlehurst, S.A. Fakoorian, W. Xu, and Ch. Zhao, "Device-to-device communications: the physical layer security advantage." in IEEE ICASSP, 2014.

[7] M. Cagalj, S. Capkun, and J.P. Hubaux, "Key agreement in peer-to-peer wireless networks," in Proc. IEEE (Special Issue on Cryptography and Security), 2006.