International Journal of Engineering Research and Reviews ISSN 2348-697X (Online)
Vol. 3, Issue 2, pp: (43-47), Month: April - June 2015, Available at: www.researchpublish.com

Secure Key Establishment for Device-To-Device Communications among Mobile Devices

[1]Waleed Aldosari, [2] Tarik El Taeib

Computer science and engineering University of Bridgeport Bridgeport, USA

***Abstract*: In recent years Device-to-Device (D2D) communications have become an attractive solution for enhancing the performance of traditional cellular networks, although the relevant security issues involved in D2D communications have not been fully addressed yet. In this work, I analyzed security requirements and challenges for D2D communications, and simulated a secure and efficient key agreement protocol, which enables two mobile devices to establish a shared secret key for D2D communications without prior knowledge based on the Diffie-Hellman key agreement protocol and commitment schemes. I present the design details and security analysis of the proposed protocol along with simulations.**

***Keywords:* D2D Communications, MITMA, Diffie-Hellman, Commitment Schemes.**

I. INTRODUCTION

The emergence and popularity of personal mobile devices, such as smartphones and tablets, generates large amount of data traffic by accessing the Internet and downloading applications, which imposes a huge burden for the cellular infrastructure and spectrum. [1]

Device-to-Device (D2D) communications have been introduced to offload the traffic burden from cellular infrastructure to personal devices. The D2D technology enables mobile device users directly establish wireless links between each other, without passing through the public cellular infrastructure or access points. [1]

Due to the broadcast nature of wireless communication, wireless channels are considered vulnerable to a variety of attacks, and security is one of the major concerns for D2D communications. To secure the communication between two end users of a D2D link, establishing a shared secret key is the first and most significant step. However, lack of trusted third party and infrastructure under D2D connection environment makes this step a non-trivial task. The well-known Diffie-Hellman key agreement protocol enables two parties jointly.

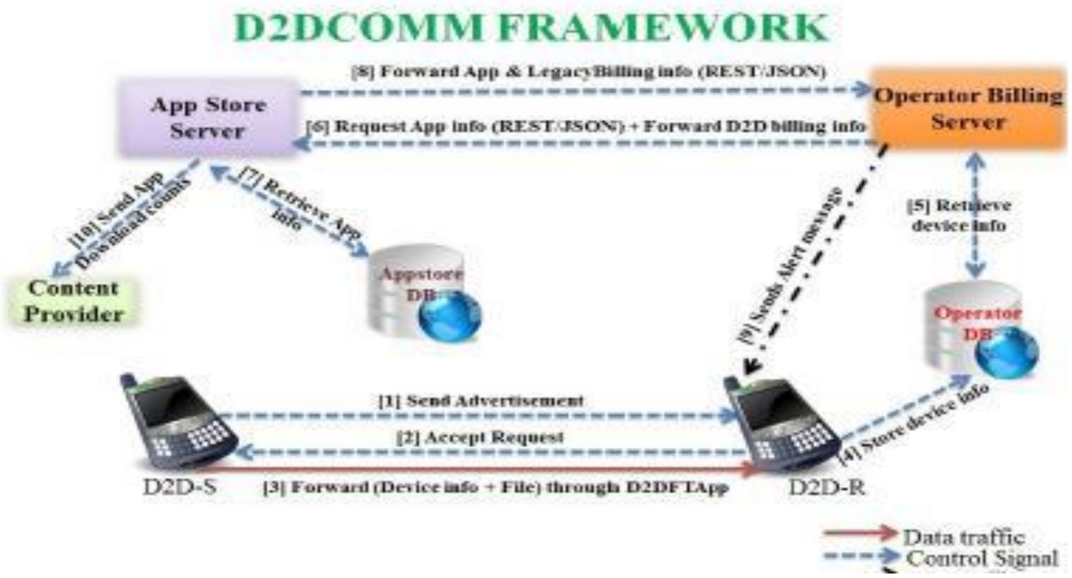

Figure 1. D2D Scenerio

Establish a shared secret key without any prior knowledge. However, this protocol is vulnerable to the man-in-the-middle attack (MITMA): an active adversary makes independent connections with the victims, making them believe that they are talking directly to each other. To address this issue, researchers have come up with various Diffie-Hellman based cryptographic protocols, which can prevent the MITMA by conducting mutual authentication. [2]

One simple protocol is, in which devices A and B exchange the hashes of their public keys over a secure channel, thus performing the mutual authentication. However, this protocol requires a large number of bits to be mutually authenticated. The MANA protocol reduces the size of the authentication message to k bits, but requires a stronger notation of authentication channel. [1, 2]

The report is organized as follows. In Section II, a brief overview of security challenges of D2D communications along with problem formulation discussed, in Section III a short overview of Diffie-Hellman and Commitment Schemes is given. In Section IV, a MATLAB simulation based on Diffie-Hellman and Commitment Schemes is presented, which describes the basic concepts of establishing a shared secret key. Finally, the results of simulations and concluding remarks are given in Section V and Section VI respectively.

II. PROBLEM FORMULATION

Security is one of the major concerns that need to be well addressed before D2D technique gets widely accepted and implemented. Due to the broadcast nature of wireless channels, wireless communication such as Wi-Fi and Bluetooth is vulnerable to a variety of attacks that challenges the three basic principles of security which are confidentiality, integrity and availability.

Some common attack vectors include surreptitious eavesdropping, message modification and node impersonation. For example, by stealthy listening to the communication between two devices, an attacker can gain critical or private information, such as trade secrets or identity related information. Thus, the D2D communications between devices need to be properly secured which in fact can be done via cryptography solutions that are needed to encrypt the messages while they are transmitted via wireless channels. Numerous encryption algorithms have been well developed which can provide different security levels for the encrypted messages, but all of them require two devices agree on a shared secret (either a shared secret key or each other's public keys). [1]

Preloading secure keys into mobile devices is neither efficient nor practical because of the large number of mobile devices, the diversity of device manufacturers and lack of standards. Besides, a trusted third party or infrastructure is not likely to be available in the D2D mobile environment. Consequently, how to establish a shared secret between devices is one of the main challenges for secure D2D communications. [1]

One of the easiest and straightforward way to establish a shared secret between two devices is that the two end users of the D2D link interactively set up a secret key via human negotiation (such as making a phone call if they are in distance). The problem for this is that the shared secret established by human interaction will be too weak in most cases so the attackers do not even need to be smart to crack this weak secret via brute force method, considering current computation power. To deal with this issue, cryptologists and researchers come up with two types of approaches which enable two individuals to establish a secure enough secret key: Diffie-Hellman key establishment protocol and secret key extraction from physical channel characteristics. [1]

Physical layer based secret key generation methods have been proposed in recent years as alternative solutions for traditional Diffie-Hellman key agreement protocol. Unlike Diffie- Hellman key agreement protocol, whose security is guaranteed by the computational hardness of discrete logarithms, these physical layer based methods rely on the randomness and uniqueness of wireless fading channel properties: temporal variation, spatial variation and reciprocity. [1]

Generally, the two devices first send channel probing packets to measure the physical metrics of the wireless channel, then after using quantization and error correction technique, these two devices can yield the same secret key. The main problem for this type of methods is that the secret key generation rate is in most case very low. Users have to send lots of channel probing packets to achieve a secret key with enough bits and randomness. The communication overhead and relatively longer key generation time are not quite desirable for the case of D2D communications.

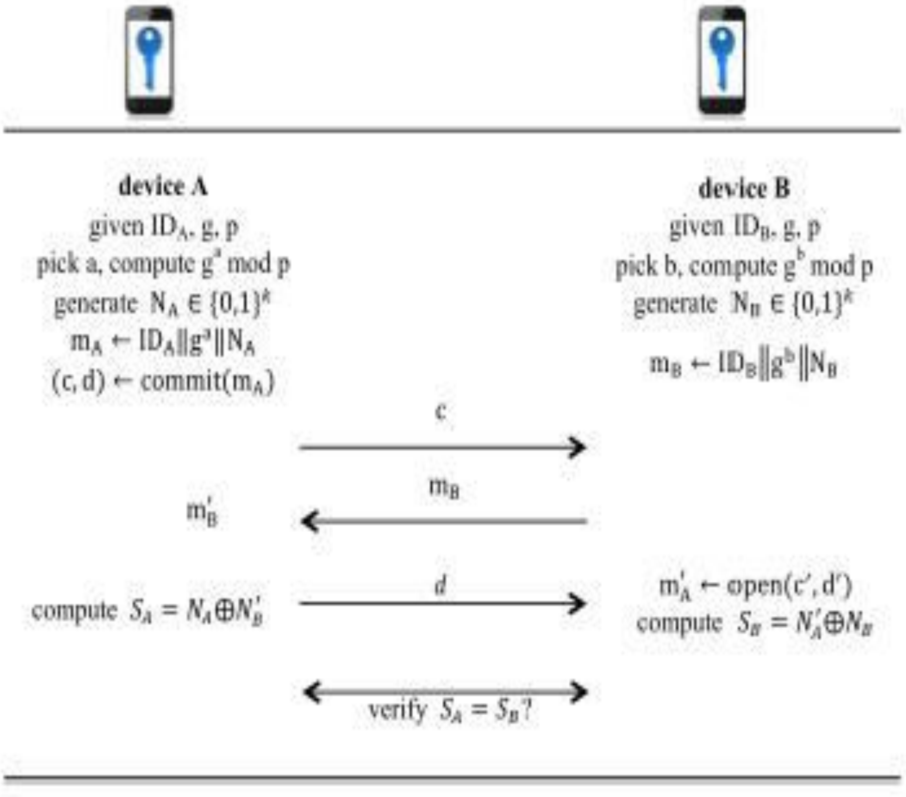
device A
given ID_A, g, p
pick a, compute g^a mod p
generate $N_A \in \{0,1\}^k$
$m_A \leftarrow ID_A \| g^a \| N_A$
$(c, d) \leftarrow commit(m_A)$
device B
given ID_B, g, p
pick b, compute g^b mod p
generate $N_B \in \{0,1\}^k$
$m_B \leftarrow ID_B \| g^b \| N_B$
c
m_B
m'_B
d
compute $S_A = N_A \oplus N'_B$
$m'_A \leftarrow open(c', d')$
compute $S_B = N'_A \oplus N_B$
verify $S_A = S_B$?

Figure 2. Secure Key Exchange Protocol

Diffie-Hellman cryptosystem is the oldest public key system still in use, which allows two individuals to agree on a shared secret key, even though they can only exchange messages over public channels and work as follows:

Assume p and q are publicly known to two devices A and B (if not, A can put them into its message and send it to B), A and B both randomly generate value a and b. A computes $g^a\ mod\ p$ and sends it to B, correspondingly, B computes $g^b\ mod\ p$ then sends it to A. At the last stage, A computes $= (g^b)^a\ mod\ p$, B computes $s = (g^a)^b\ mod\ p$. Both A and B will arrive at the same value, since $(g^b)^a$ and $(g^a)^b$are equal $mod\ p$. $(g^a)^b\ mod\ p$ will be the established shared secret between A and B, thus can be subsequently used as encryption key for future communication. The implementation of Diffie-Hellman key agreement protocol requires some extent of computation capacity, since p, a and b can be quite large numbers. However, mainstream mobile devices on today's market have achieved gigahertz level processor frequency, so generating a secure enough shared secret, say, 156 bits, can be conducted within seconds. [1]

As is well known, the above Diffie-Hellman key agreement protocol is vulnerable to the so called the man-in-the-middle attack. Since g^a and g^b are transmitted over the public channel, there is no way for device A to know for sure whether g^b comes from device B, vice versa. Devices A will establish a shared secret with whoever transmits g^b, and it certainly might not be device B. The essential reason that the MITMA is possible is that there is no mutual authentication between these two devices. To provide the desired authentication, one intuitive solution is both devices put the obtained secret key to a one-way hash function, e.g. MD5, to generate a hash value h(K), then compare the hash value via a trusted channel (for example, output the computed hash code on device screens and perform visual or verbal comparison). If the mutual authentication process agrees, then both devices can confirm that they have established a shared secret key with each other. [1] [2]

The main issue about the above mutual authentication procedure is that the number of bits needed to be checked by the user is too large. The output of a hash function is usually over 128 bits (32 hexadecimal digits), and visually or verbally checking them is a non-trivial task. Using truncation of the hash code can drastically reduce the number of digits needed to be checked, but doing this will introduce serious security weakness. [1] [2]

III. OVERVIEW OF USED ALGORITHMS

A. Problem Formulation and Assumptions:

Two mobile device users equipped with a smartphone or tablet which is capable of communicating over a wireless channel, want to establish a shared secret key for their D2D communications. Both devices have the computation capacity to perform Diffie-Hellman key agreement protocol, and are capable of displaying sequence of digits. These two users do not have any pre-shared cryptographic information, and there is no trusted third party or infrastructure available. They can visually or verbally recognize each other for the purpose of mutually authenticate a short message.

Presumably devices A and B agree on a finite cyclic group $\mathbb{G}$, its generating element $\boldsymbol{g}$, and a large prime number $\boldsymbol{p}$. We assume $\mathbb{G}$ to be a subgroup of $\mathbb{Z}_{\boldsymbol{p}}^{*}$ of prime order $\boldsymbol{q}$, where, $\mathbb{Z}_{\boldsymbol{p}}^{*}$ is the multiplicative group consists of nonzero integers modulo $\boldsymbol{p}$.

The attacker has fully control over the wireless channel. It can overhear, intercept, and modify any message. The attacker can also initiate a conversation with any other user.

B. Diffie-Hellman Algorithm:

Suppose that Alice and Bob have not met yet nor exchanged keys, but they want to establish a shared secret key $\boldsymbol{K}$ by exchanging messages over an unsecured channel. First Alice and Bob agree on a large prime number $\boldsymbol{p}$ and a generator $\boldsymbol{a}$. These need not be kept secret, so Alice and Bob can agree over an unsecured channel. Then the protocol proceeds as follows,

1. Alice chooses a random (large) $\boldsymbol{x}$ and computes the least positive residue $\boldsymbol{X}$ of $\boldsymbol{a^x}\ \boldsymbol{modulo}\ \boldsymbol{p}$ and then send $\boldsymbol{X}$ to Bob (and keeps $\boldsymbol{x}$ secret)

2. Bob chooses a random (large) $\boldsymbol{y}$ and computes the least positive residue $\boldsymbol{Y}$ of $\boldsymbol{a^y}\ \boldsymbol{modulo}\ \boldsymbol{p}$ and then sends $\boldsymbol{Y}$ to Alice (and keeps $\boldsymbol{y}$ secret)

3. Alice computes the least positive residue of $\boldsymbol{Y^x}\ \boldsymbol{modulo}\ \boldsymbol{p}$, and Bob computes the least positive residue of $\boldsymbol{X^y}\ \boldsymbol{modulo}\ \boldsymbol{p}$.

Since$\boldsymbol{Y^x} = (\boldsymbol{a^y})^{\boldsymbol{x}} = (\boldsymbol{a^x})^{\boldsymbol{y}} = \boldsymbol{X^y}\ \boldsymbol{mod}\ \boldsymbol{p}$ they have a shared secret key . In Diffie-Hellman Protocol, $\boldsymbol{K}$ is the shared secret key independently generated by both Alice and Bob. The key exchange is complete, since Alice and Bob are agreement of $\boldsymbol{K}$

A cryptanalyst, Eve, listening to the channel would know $\boldsymbol{p}, \boldsymbol{a}, \boldsymbol{X}$and $\boldsymbol{Y}$ but neither $\boldsymbol{x}$ nor $\boldsymbol{y}$. Thus Eve faces what is called as Diffie-Hellman Problem. If Eve can solve DLP-Discrete Log Problem, then she can clearly solve the DHP-Diffie-Hellman Problem. Whether the converse is true or not is unknown. In other words, it is not known if it is possible for a cryptanalyst to solve the DHP without solving DLP. Nevertheless the consensus is that the two problems are equivalent. Thus for practical purposes one may assume that the Diffie-Hellman Key Exchange Protocol is secure as long as DLP is intractable.

C. Commitment Schemes:

A commitment scheme allows one user to commit to a chosen value or statement while keep it hidden to others, with the ability to reveal the commitment value latter. A commitment scheme has the following two main properties:

1) a user cannot modify the value or statement after they have committed to it; that is, the commitment scheme is binding and

2) the receiver can only know the committed value after the sender "opens" it; that is, the commitment scheme is hiding. A commitment scheme is defined by two algorithms **Commit** and **Open**:

Commit: $(\boldsymbol{c}, \boldsymbol{d}) \leftarrow \boldsymbol{m}$ m transforms a value $\boldsymbol{m}$ into a commitment/open pair $(\boldsymbol{c}, \boldsymbol{d})$ The commit value c reveals no information of $\boldsymbol{m}$, but with decommit value $\boldsymbol{d}$ together $(\boldsymbol{c}, \boldsymbol{d})$ will reveal $\boldsymbol{m}$.

Open: $(\boldsymbol{c}, \boldsymbol{d}) \rightarrow \boldsymbol{m}$ output original value $\boldsymbol{m}$ if $(\boldsymbol{c}, \boldsymbol{d})$ is the commitment/open pare generated by Commit($\boldsymbol{m}$).

D. Protocol Analysis:

Here design of the key agreement protocol, which is based on the traditional Diffie-Hellman key agreement protocol and a commitment scheme, is presented. In this protocol, two mobile users A and B respectively generate $\boldsymbol{k}$-bit random strings $\boldsymbol{N_A}$ and$\boldsymbol{N_B}$, and $\boldsymbol{N_A \oplus N_B}$ is the short authentication string for mutual authentication.

At the initial stage, user A and B select their Diffie-Hellman parameter $\boldsymbol{a}$ and $\boldsymbol{b}$, then compute $\boldsymbol{g^a}$ and $\boldsymbol{g^b}$. A and B randomly generate their $\boldsymbol{k}$ -bit strings $\boldsymbol{N_A}$ and $\boldsymbol{N_B}$. Then $\boldsymbol{m_A = ID_A \| g^a \| N_A}$ and $\boldsymbol{m_B = ID_B \| g^b \| N_B}$ are formed by concatenation, in which $\boldsymbol{ID_A}$ and $\boldsymbol{ID_B}$ are human readable identifiers for user A and B, such as names or e-mail addresses. A also needs to calculates the commitment/opening $(\boldsymbol{c}, \boldsymbol{d})$ for $\boldsymbol{m_A = ID_A \| g^a \| N_A}$.

After the initial stage, user A and user B perform the following message exchange over their D2D communications channel. User A sends the $\boldsymbol{c}$, the commitment value of $\boldsymbol{m_A}$ to user B; after receiving $\boldsymbol{c}$, user B sends $\boldsymbol{m_B}$ to user A. In return, user A sends the decommit value $\boldsymbol{d}$ to user B. User B opens the commitment and gets $\boldsymbol{m_A = ID_A \| g^a \| N_A}$

In the final stage, user A and B generate the $\boldsymbol{k}$bits authentication string by $\boldsymbol{S_A = N_A \oplus N'_B}$and $\boldsymbol{S_B = N'_A \oplus N_B}$, in which $\boldsymbol{N'_B}$ and $\boldsymbol{N'_A}$ are derived from messages received by A and B. Then user A and B verify if $\boldsymbol{S_A = S_B}$ via trusted channel (visual or verbal comparison). If the authentication strings match, A and B accept each other's Diffie-Hellman parameters and calculate the shared secret key $\boldsymbol{K = (g^a)^b \ mod\ p}$. The reason for comparing authentication string before generating Diffie-Hellman secret key is that if the strings do not match, both users can save the computation for secret key generation.

A. Simulation Model and Scenarios:

We used two mobile devices A and B during simulations and created all numbers and functions on their side separately while transactions taken place under user control. The simulation codes are user friendly and also prompts which allows to track the process at each step both for Diffie-Helman and Commitment Schemes.

IV. RESULTS

Since cryptographic simulations are not in the form of the familiar table and graph outputs, this part does not contain any, however we tested both Diffie-Hellman algorithm and Commitment Schemes along with the protocol analyzed with different values of parameters, including different Diffie-Hellman parameters a and b, and modulo prime numbers as well as different $\boldsymbol{k}$ -bit strings for commitment schemes. Although simulation codes check and validate some of the inputs like prime input or not, it is not for complete implementation of the routines in a tablet, mobile or similar device, so as long as input parameter properties are satisfied, a user can test different entries for both Diffie-Hellman and Commitment Schemes. Apart from Matlab 52-digit limitation(which is known and provides enough precision arithmetic for simulation purposes) we did not face any problem in calculations, and took successful results in all test cases where for commitment schemes it is also provided as screen output as whether Authentication takes place or not.

V. CONCLUSION

In this work, we analyzed the security requirements and challenges for secret key establishment between two mobile devices using Diffie-Hellman algorithm and Commitment Schemes as basic models. The simulation results shows that both protocols can be used efficiently for D2D communications, and can be used effectively.

REFERENCES

[1] W.Shen, W.Hong, X.Cao, B.Yin, D.M.Shila, Y.Cheng "Secure Key Establishment for Device-to-Device Communications", Illinois Institute of Technology, 2013

[2] S. Ramasubramanian, S. Chung, L.Ding, "Secure and Smart Media Sharing Based on Direct Communications Among Mobile Devices Underlying in LTE-A Cellular Network", University of Washington 2013

[3] Matlab Users Guide, The Mathworks Inc.