Delay and Cost Performance Analysis of the Diffie-Hellman Key Exchange Protocol in Opportunistic Mobile Networks

B Soelistijanto^1 , V Muliadi^2

Department of Informatics, Faculty of Science and Technology, Sanata Dharma University, Yogyakarta, Indonesia

Email: ^{1} b.soelistijanto@usd.ac.id, ^{2}vmuliadi@max-metal.us

Abstract. Diffie-Hellman (DH) provides an efficient key exchange system by reducing the number of cryptographic keys distributed in the network. In this method, a node broadcasts a single public key to all nodes in the network, and in turn each peer uses this key to establish a shared secret key which then can be utilized to encrypt and decrypt traffic between the peer and the given node. In this paper, we evaluate the key transfer delay and cost performance of DH in opportunistic mobile networks, a specific scenario of MANETs where complete end-to-end paths rarely exist between sources and destinations; consequently, the end-to-end delays in these networks are much greater than typical MANETs. Simulation results, driven by a random node movement model and real human mobility traces, showed that DH outperforms a typical key distribution scheme based on the RSA algorithm in terms of key transfer delay, measured by average key convergence time; however, DH performs as well as the benchmark in terms of key transfer cost, evaluated by total key (copies) forwards.

1. Introduction

The subject of key distribution is one of the important issues in key management [1]. An obvious method of distribution of keys is simply by hand. This method was frequently used in the days of couriers. However, it is used only infrequently nowadays, since most key distribution is performed automatically. Automatic distribution is not only more convenient, but often even essential; for instance, in telephone or computer networks, which require two parties to transmit their security keys along the same communication line. In these cases often two types of keys are employed: keys which are used for the actual security of the data (so-called *session keys*), and keys which are used for the security of these session keys during transmission (so-called *transportation keys*).

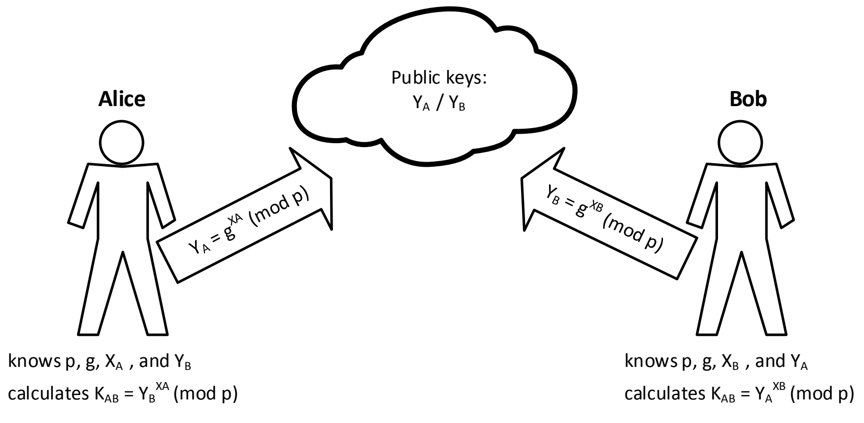
The Diffie-Hellman (DH) protocol [2] is a method for exchanging keys in the network. In this algorithm, two parties unknown to one another can set up a private however arbitrary key for their symmetric key cryptosystem. Along these lines, there is no requirement for Alice and Bob to meet ahead of time, or utilize a safe dispatch, or utilize some other secret means, to choose a key. DH was the first practical method for setting up a “shared secret” over an unsecured communication channel. The security of this algorithm is based on the hardness of a certain computational problem. This paper discusses the delay and cost performance analysis of the DH key exchange protocol in opportunistic mobile networks (OMNs), a class of delay-tolerant networks (DTNs) [3] where nodes come into contact with each other at unpredictable intervals and the duration of each contact is also unpredictable. The area of key management in DTNs is relatively new and many research challenges remain to date [4]. Traditional key management is not suitable for DTNs due to the environment limitations and technical constraints, e.g. long round trip delays and frequent link disconnections. The author in [5] identified some requirements for key management in DTNs. The works in [6,7,8] proposed new solutions to address the security issues of key management in DTNs.

In this paper, however, we discuss the key transfer delay and cost performance analysis of DH in OMNs. To best of our knowledge, the performance evaluation of DH in OMNs in terms of key transfer delay and cost has not been discussed before. In delay-tolerant networks, such as OMNs, where end-to-end delay is very large, key transfer delay is an important aspect that directly affects the performance of DH. Indeed, a low key transfer latency is required to achieve high performance of DH. In this analysis, we consider average key convergence time to identify the key transfer delay performance of DH in OMNs. Besides transfer delay, delivery cost is an important parameter in mobile communication networks, since mobile devices typically have limited resources, e.g. storage and power. Furthermore, we use total key (copies) forwards as a metric to quantify the key transfer cost of DH in OMNs.

The rest of the paper is structured as follows. Section 2 gives an overview of the DH key exchange algorithm. Section 3 describes our experimental setup for evaluating the key transfer delay and cost performance of DH in OMNs. A performance analysis of the protocol compared to a typical key distribution scheme based on the RSA algorithm is reported in Section 4. Finally, Section 5 concludes the paper.

2. The Diffie-Hellman Key Exchange Protocol

The Diffie-Hellman (DH) protocol offers an efficient method for exchanging keys in the network. Consider a network of n nodes (parties) who communicate bilaterally in an enciphered form, controlled by a key. In the network, the total number of keys is $n(n-1)$ in which two nodes are involved during each communication. Each node must therefore have $(n-1)$ keys available to be able to communicate with the other $(n-1)$ nodes. This kind of communication will require a large number of keys and managing these keys correctly can be very complicated.



Public keys:
Y_A / Y_B
Alice
Bob
$Y_A = g^{XA}$ (mod p)
$Y_B = g^{XB}$ (mod p)
knows p, g, X_A , and Y_B
calculates $K_{AB} = Y_B^{XA}$ (mod p)
knows p, g, X_B , and Y_A
calculates $K_{AB} = Y_A^{XB}$ (mod p)

Figure 1. The Diffie-Hellman key exchange protocol

Diffie and Hellman (1976) have devised a method with which the number of keys can be reduced drastically. Their system is based on the exponential and logarithmic functions. Given that all nodes in the network select the same prime number p and a base g. Each node N_i may now choose a secret key x_i and calculates y_i according to $y_i = g^{x_i} \pmod{p}$; the value of y_i is then made public. Suppose a node N_i wishes to communicate with a node N_j. This will require a mutual session key K_{ij}, with which the message can be enciphered and deciphered. Node N_i must first calculate $K_{ij} = {y_j}^{x_i} \pmod{p}$. N_i can perform this calculation since p is published, x_i is N_i's own key, and y_j was disclosed to N_i by N_j.

On the other hand, node N_j can calculate the session key $K_{ji} = {y_i}^{x_j} \pmod p$. This will result in exactly the same session key for both nodes ($K_{ij} = K_{ji}$), due to symmetry of the power function. In figure 1, we illustrate the DH key exchange protocol between two parties, Alice and Bob.

3. Simulation Setup for Evaluating the Diffie-Hellman Protocol in OMNs

In this section, we discuss the choice of simulation scenarios, evaluation metrics, and benchmark protocols to evaluate the delay and cost performance of DH in OMNS. We implement DH using the ONE simulator [9], an event-driven simulator for mobile opportunistic networks. In our simulations, the number of nodes and the length of simulation time vary depending on the node mobility scenarios. The node buffer size and the message size are set to 10 MB and 4 kB, respectively. Moreover, we assume that a message can only carry a single key. The simulations were run 10 times for both DH and its protocol benchmark with different random number seeds.

For the simulation's node mobility scenario, we use a random movement model and real human mobility traces. For the latter case, we exploit the Infocomm [10], Reality [11] and Sassy [12] human contact datasets. In Infocomm, 41 iMote Bluetooth-enabled devices were distributed to attendees at the IEEE Infocomm conference in Miami in 2005. These devices recorded the human contacts occurred during the 3-day seminar. In Reality, on the other hand, 100 smart phones were deployed among the students and staffs of MIT over period of 9 months. These phones were running software that logged contacts with other phones, capturing academic activities in the campus over an academic year. The Sassy trace, in contrast, was collected using a mobile sensor network with TMote invent devices carried by 25 participants from the University of St. Andrews for period of 74 days.

For performance analysis, we use two evaluation metrics as follows:

1. **Average key convergence time**: the mean of the times required before all the nodes in the network can reach knowledge regarding the latest updated of the key of a particular node.
2. **Total key (copies) forwards**: the total number of key (copies) forwarded during node contacts throughout the simulation time.

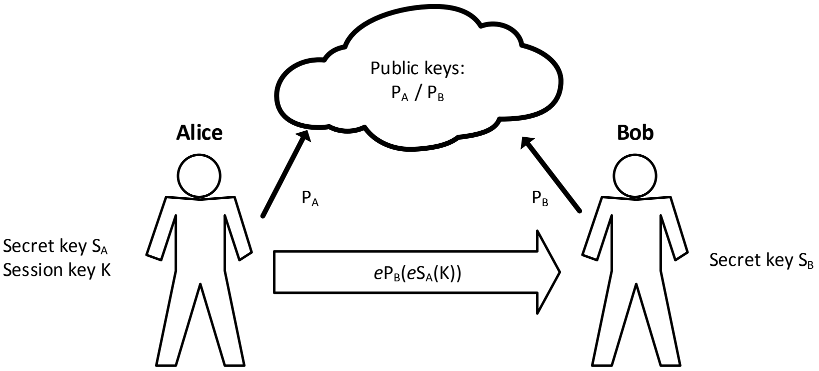


Figure 2. A key distribution scheme based on the RSA algorithm

In order to benchmark the performance of DH, we consider a typical key distribution scheme based on the RSA algorithm (hereafter called RSA) as follows. In figure 2, consider two parties, Alice and Bob, whose public keys P_A and P_B, respectively, are made available in the network, and secret keys are denoted by S_A and S_B. Suppose that Alice wishes to transfer a (session) key K to Bob. Alice may do this by first enciphering the key K with her own secret key and subsequently enciphering this result with Bob's public key. The total result can be written as $(K^{S_A})^{P_B}$. Then, Bob can find the original key K by decrypting the received message with his own secret key followed by a second decipherment with Alice's public key.

In our simulations, a new public key is generated at a randomly selected node at a rate of one key per hour for both DH and RSA. Moreover, in RSA, we assume that after a node broadcast its new public key to all nodes in the network, the node promptly sends an individual encrypted session key K to each the network node. Therefore, the average key convergence time in RSA is calculated as the mean of the times when all the network nodes are able to find the original key K of a node by decrypting the received messages minus the time when K is created in the original node. Furthermore, the peer node is able to disclose the encrypted session key K whenever it has knowledge of the latest updated public key of the original node. This is however not the case in DH since K is never transported over the network but is calculated autonomously on both communicating end nodes, providing that each end node has information of the public key of the other end node (as shown in figure 1). As a result, the key convergence time in DH can be calculated as the average of times when all nodes in the network receive a particular node' public key minus the time when the key is created in the original node.

Finally, to broadcast nodes' public keys in the network (in DH and RSA) and to deliver session keys K to the particular destination (in RSA) we use Epidemic routing [13]. The algorithm is flooding-based in nature, as nodes continuously replicate and transmit messages to newly discovered contacts that do not already possess a copy of the message. Epidemic routing results in a high per message delivery probability and the lowest delay. Despite its benefits, this oblivious forwarding strategy consumes much the constraint resources of mobile nodes, such as storage and power.

4. Simulation Results

In this section, we present the simulation results that compare the key transfer delay and cost performance of DH with that of RSA in OMNs. In the first experiment, we consider a random node movement model in the simulation. In the second experiment, we use real human contact data traces as the simulation's node mobility scenario.

4.1. Random Node Movement Scenario

We now discuss the key transfer delay and cost performance comparison results between DH and RSA in a random node movement scenario. We used a synthetic random walk model available in the ONE simulator's library. In this setting, node contacts are spread evenly between the nodes over the network. From the simulation results, in figure 3 we depict the key transfer delay and cost performance comparison results of DH and RSA in this random scenario. We defined a fixed simulation area at 2500m x 2500m but varied the number of nodes in the network from 50 up to 150 nodes. The length of the simulation time was 2500ks for both protocols.

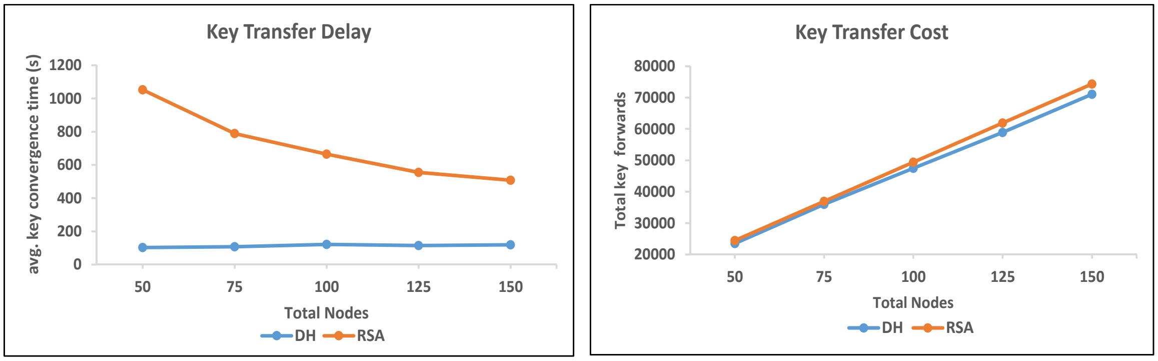


Figure 3. Key transfer delay and cost performance comparison of DH and RSA in the random node mobility scenario

Figure 3 shows that DH outperforms RSA in terms of key transfer delay, measured by average key convergence time. This is because the key convergence time in DH relies merely on the speed of public keys spreading in the network. In RSA, on the other hand, the key convergence time depends not only on the broadcast rate of public keys to all the network nodes, but also on the delivery time of session keys to their destinations (unicast transmissions). Moreover, the increase of total nodes in the network significantly reduces the key transfer delay in RSA, but it gives little impact in DH. In terms of key transfer cost performance, on the other hand, both protocols perform almost similarly. Indeed, DH slightly reduces the total key (copies) forwards of RSA. In RSA, the majority of key (copies) distributed in the network are nodes' public keys and the delivery of session keys consequently less contributes on the total key (copies) forwarded during the simulation time. In line with this, the absence of the unicast transmissions of session keys in DH slightly decreases the key transfer cost below that of RSA. Finally, the key transfer cost grows linearly with the increasing of total nodes in the network for both DH and RSA.

4.2. Real Human Mobility Scenario

In the second experiment, we consider a real human mobility scenario to evaluate the key transfer delay and cost performance of DH compared to that of RSA. We used Reality, Sassy and Infocomm as the simulation's node mobility scenario. We again used the simulation settings of the previous experiment, except for the number of nodes and the length of simulation time which vary depending on the datasets.

From the simulation results, in figure 4 we depict the key transfer delay performance comparison of DH and RSA, measured by avg. key convergence time, for all the datasets. It is clear that DH outperforms RSA in this performance metric (i.e. a lower avg. key convergence time of DH compared to that of RSA) in all there datasets. Moreover, the key transfer delay performance's difference between DH and RSA is more obvious in Reality and Sassy. In these datasets, node contacts are sparser (than Infocomm's) and, as a result the delivery time of session keys to the destinations is larger, leading to the significant increase of the key transfer delay of RSA beyond that of DH in both Reality and Sassy.

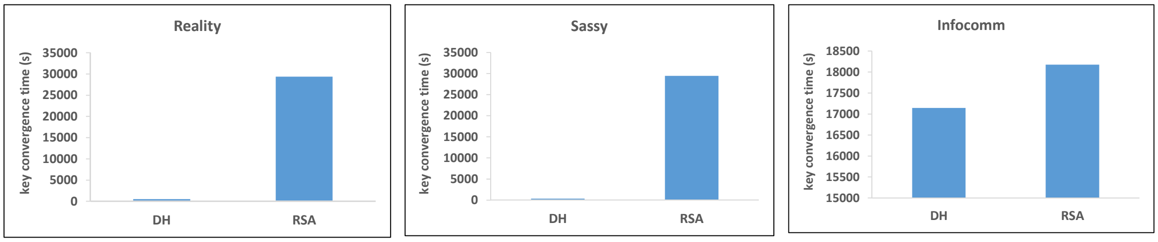


Figure 4. Key transfer delay performance comparison of DH and RSA in the real human mobility scenario

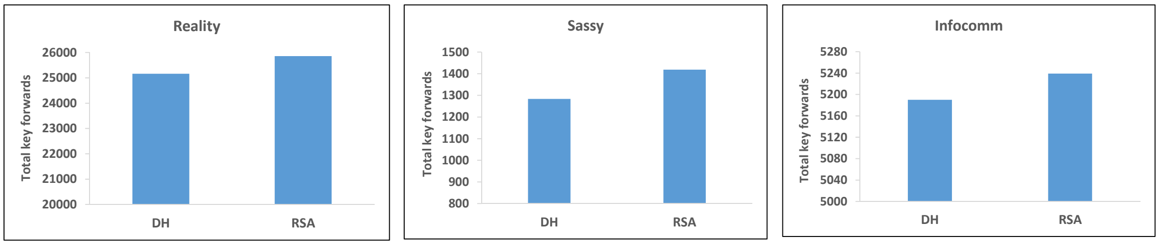


Figure 5. Key transfer cost performance comparison of DH and RSA in the real human mobility scenario

Finally, in figure 5 we show the key transfer cost performance comparison results, measured by total key (copies) forwards, for DH and RSA in all the datasets. As in the previous scenario, both protocols perform nearly equally in this evaluation metric and DH slightly reduces the total key (copies) forwards of RSA in all three datasets. The explanation of this is similar to that given in the key transfer cost performance evaluation in the previous scenario as follows: in RSA, the unicast transmissions of session keys give a little impact on the total key (copies) forwarded during the simulation time since the majority of the key (copies) distributed in the network are nodes' public keys. According this, the absence of the unicast transmissions of session keys in DH results in a small decrease of the key transfer cost below that of RSA in all the datasets.

5. Conclusion

We have discussed the delay and cost performance evaluation of the DH key exchange protocol in OMNs. We have demonstrated that DH outperforms a typical key distribution scheme based on RSA in terms of key transfer delay, measured by average key convergence time, in both the random node mobility and real human mobility scenarios. However, in the key transfer cost performance DH performs as well as RSA in both scenarios. Indeed, DH slightly reduces the total key (copies) forwards below that of RSA.

References

[1] Van der Lubbe J C A 1998 *Basic Methods of Cryptography* (London: Cambridge University Press) chapter 8 pp 194-199

[2] Diffie W and Hellman M E 1976 New Directions in Cryptography *IEEE Trans. on Information Theory* vol IT-22 no 6 (New Jersey: IEEE) pp 644-650

[3] Fall K 2003 A Delay-Tolerant Network Architecture for Challenged Internets *Proc. ACM SIGCOMM* (Karlsruhe: ACM) pp 27-34

[4] Clarke N L, Katos V, Menesidou S-A, Ghita B, and Furnell S 2012 A Novel Security Architecture for a Space-Data DTN *Proc. Int. Conf. on Wired/Wireless Internet Communications* (Santorini: Springer) pp 342-349

[5] Menesidou S A, Katos V, and Kambourakis G 2017 Opportunistic Key Management in Delay-Tolerant Networks *Int. J. Information and Computer Security* vol 9 no 3 (Geneva:Inderscience) pp 212-228

[6] Menesidou S A and Katos V 2012 Authenticated Key Exchange (AKE) in Delay-Tolerant Networks *Proc. Int. Conf. on Information Security* (Creta: Springer) pp 49-60

[7] Andrade D and Albini L C P 2016 Fully Distributed Public Key Management through Digital Signature Chains for Delay and Disrupt Tolerant Networks *Proc. Int. Conf. on Mobile Ad Hoc and Sensor Systems* (Brasilia: IEEE) pp 316-324

[8] Lv X, Mu Y, and Li H 2014 Non-Interactive Key Establishment for Bundle Security Protocol of Space DTNs *IEEE Trans. on Information Forensics and Security* vol 9 no 1 (New Jersey: IEEE) pp 5-13

[9] Keranen A, Ott J, and Karkkainen T 2009 The ONE Simulator for DTN Protocol Evaluation *Proc. Int. Conf. on Simulation Tools and Techniques* (Rome: IEEE) pp 1-10

[10] Scott J, Gass R, Crowcroft J, Hui P, Diot C, and Chaintreau A 2009 CRAWDAD Dataset Cambridge/Haggle/Infocom (online: http://crawdad.cs. dartmouth. edu/cambridge/haggle/imote/infocom)

[11] Eagle N and Pentland A 2006 Reality Mining: Sensing Complex Social Systems *J. Personal and Ubiquitous Computing* vol 10 no 4 (London: Springer-Verlag) pp 255-268

[12] Bigwood G, Henderson T, Rehunathan D, Bateman M, and Bhatti S 2011 CRAWDAD Dataset st_andrew/sassy v. 2011-06-03 (online: http://crawdad.org/st_andrews/sassy/20110603/)

[13] Vahdat A and Becker D 2000 Epidemic Routing for Patially Connected Ad Hoc Networks *Tech. Report CS-200006 Duke Univ.* (Durham: Duke University)