A STUDY ON DIFFIE-HELLMAN KEY EXCHANGE PROTOCOLS

Manoj Ranjan Mishra¹, Jayaprakash Kar^{2 §}

¹School of Computer Application
KIIT University, Bhubaneswar, INDIA

²Department of Computer Science and Engineering
The LNM Institute of Information Technology
Jaipur (Raj), INDIA

Abstract: Securing network traffic has always been a must requirement for any network application that employs insecure communication channel. The reason is to provide protection for the transmitted data over the network against unauthorized disclosure and modification of the messages between communicating parties. A Key exchange protocol is the cryptographic primitive that can establish a secure communication. The first Key exchange protocol was introduced by Diffie-Hellman. The purpose of the Diffie-Hellman protocol is to enable two parties to securely exchange a session key which can then be used for next symmetric encryption of messages. However, Diffie-Hellman itself does not authenticate the communicating entities. In this paper, we study on Diffie-Hellman Key exchange protocol. Subsequently describe authenticated key exchange protocol and One-pass key exchange protocol, which are the variants of Diffie-Hellman protocol.

Key Words: authenticated key exchange, key compromise impersonating attack, ephemeral key compromise attack

1. Introduction

The need for a key exchange protocol over an insecure communication channel

Received: May 5, 2016

Revised: April 28, 2017

Published: April 28, 2017

© 2017 Academic Publications, Ltd.

url: www.acadpubl.eu

[§]Correspondence author

is raised to prevent unauthorized access or accidental disclosure of the information while transmission process between entities over a network. Communicating between two parties on a public network needs to be secure to prevent any attempt from attackers to read transmitted messages. Secure transmission means encrypting the message with an encryption key and then sending it from one party to another. The problem is how to deliver the key between those two parties securely. The answer is to use key exchange protocols which verify identities of each party to another, create and distribute the key among them securely. Key exchange protocols can be categorized into two categories [1]: key transport protocols and key agreement protocols. In key transport protocols, the session key is first created by a member of communicating parties and then transmitted securely to the other. In other hands, the key agreement protocol relies on some information from the both parties to derive the session key from. One of the first key exchange protocols appeared is called Diffie-Hellman key exchange by Diffie and Hellman [2]. The purpose of Diffie-Hellman protocol is to enable two parties to securely exchange a session key which can then be used for next symmetric encryption of messages. The idea of Diffie-Hellman protocol is to calculate a session key by the communicating entities based on public parameters that are shared in the initial phase. This type of protocol is called key agreement protocol. Diffie-Hellman's effectiveness comes from the difficulty of calculating discrete logarithms. However, the protocol can only be used for exchanging secret data without authenticating two parties. This is the reason why Diffie-Hellman is insecure against man-in-the-middle attack. The solution for this vulnerability is to use digital signature. Variants of Diffie-Hellman protocols are proposed since its introduction to overcome different issues and vulnerabilities including the aforementioned one. We can consider the key exchange or establishment protocols from two perspectives: cost/efficiency and security. Cost includes both processing and communication costs. To get low processing cost, researchers should avoid employing public key encryption schemes such as RSA, ECC and ElGamal. Security means the protocol immunity to the known attacks such as a key compromise impersonation attack (KCI), ephemeral key compromise attack (ECI), dictionary attack, etc. In this paper, we will survey on the recent authenticate key exchange protocols (AKE) and possible attacks that can threaten them.

The rest of this paper is structured as follows. First, we define the notation used throughout the paper. In section 3, we define Diffie-Hellman key exchange protocol and why it is insecure against man-in-the-middle attack. After that, we present station to station AKE protocol (STS) as one of the earliest authenticated key exchange protocols. In section 5, we present secure and efficient key

exchange protocols. Next, we describe vulnerabilities on recent key exchange protocols and their resistant to the known attacks such as Key compromise impersonation (KCI) attack and Ephemeral key compromise (EKC) attack. Finally, we conclude the survey and list the references along with the authors' bibliographies.

2. Preliminaries: Notation

Notation	Meaning
p and q	Large prime numbers of group G
g	generator of the group G and finite cyclic group of large enough order that makes Diffie-Hellman problem difficult
X_A	Static private key of Alice
X_B	Static private key of Bob
$Y_A = g^{X_A}$	Static public key of Alice
$Y_B = g^{X_B}$	Static public key of Bob
$R_A \in Z_p$	Ephemeral random integer of Alice
$R_B \in Z_p$	Ephemeral random integer of Bob
$R_D \in Z_p$	Ephemeral random integer of Darth
K_i	Session key generated by the party i
r	ephemeral private key
R	ephemeral public key, $R = rP$
$ $	concatenation symbol
T	time-stamp
$\oplus$	XOR operation
H	Hash function

3. Diffie-Hellman Key Exchange Protocol

Diffie-Hellman is one of the earliest key agreement protocols appeared by Diffie and Hellman in 1976. The purpose of Diffie-Hellman protocol is to enable two parties to securely exchange a session key which can then be used for next symmetric encryption of messages. In this protocol, there are two public numbers shared by the communicating parties: a large prime integer q and its primitive root a . Diffie-Hellman's effectiveness comes from the difficulty

of calculating discrete logarithms. However, the protocol is vulnerable to the man-in-the-middle attack [3].

Let's assume Alice and Bob want to exchange a symmetric key, and the adversary is Darth. The attack starts as follows:

- **Step1:** Darth picks two random private keys X_{D_1} and X_{D_2} to compute the public keys Y_{D_1} and Y_{D_2} , respectively.
- **Step2:** Darth intercepts a public key transmission Y_A from Alice to Bob and sends Y_{D_1} to Bob. Moreover, Darth calculates the session key $K_{D_2} = Y_A^{X_{D_2}} \bmod q$. Bob receives Y_{D_1} to calculate $K_{D_1} = Y_A^{X_{D_1}} \bmod q$.
- **Step3:** In the same way, Darth intercepts public key transmission Y_B from Bob to Alice and sends Y_{D_2} to Alice. Moreover, Darth computes the session key $K_{D_1} = Y_B^{X_{D_1}} \bmod q$. Alice receives Y_{D_2} to calculate $K_{D_2} = Y_B^{X_{D_2}} \bmod q$.

From now on, Darth can access and read or/and modify any subsequent messages between Bob and Alice. Because the session keys of both Alice and Bob are shared with Darth. Diffie–Hellman protocol does not provide authentication of the communicating parties and consequently it is vulnerable to a man-in-the-middle attack. This vulnerability can be avoided with the use of authentication mechanisms such as digital signatures and public-key certificates.

4. Authenticated Key establishment Protocol

Researchers have been working hard to develop authenticated key establishment protocols using asymmetric techniques. In 1990, Günther has developed an authentication protocol by the help of Diffie–Hellman and ElGamal protocols [4]. The protocol was not secure against forward secrecy. Another authenticated key exchange protocol based on RSA has been developed by Okamoto and Tanaka with direct and indirect versions [5]. Fiat and Shamir [6] have proposed interactive identification protocols which support identity and involve zero-knowledge ideas. More and more protocols have been developed, but they were not cost efficient due to the redundancies in communications. Moreover, these key exchange protocols do not provide keys for future communications.

In 1992, an efficient authenticated key exchange protocol has been proposed by Whitfield Diffie et al. known as station-to-station (STS) protocol [7]. In a

two-party authenticated key exchange, the legitimate parties can compute a secret key using Diffie-Hellman and then authenticating each other by exchanging their digital signatures. As a result, STS should be secure against a man-in-the-middle attack. Now we will discuss the basic setup of STS protocol. Let's assume that the setup data of Diffie-Hellman has been shared between Alice and Bob, and they know each other's public keys using digital certificates. STS protocol proceeds as follows:

- **Step1:** Alice generates a random number x and sends the exponential g^x to Bob.
- Alice $\rightarrow$ Bob : g^x
- **Step2:** Bob generates a random number y and sends the exponential g^y . Bob also computes the exchanged key $K = g^{xy}$. Bob concatenates the exponentials in this order: (g^x, g^y) , signs them using his private key X_B , and then encrypts the signature with K . He responds to both the ciphertext and his exponential g^y to Alice.
- Alice $\leftarrow$ Bob: $g^y, E_k(S_B(g^y, g^x))$
- **Step3:** Alice computes the exchanged key $K = g^{yx}$, decrypts the ciphertext, and verifies Bob's signature using Bob's public key Y_B . Alice concatenates the exponentials in this order (g^x, g^y) , signs them using her private key X_A , and then encrypts the signature with K . She responds the ciphertext to Bob.
- Alice $\rightarrow$ Bob: $E_k(S_A(g^x, g^y))$
- **Step4:** Bob decrypts and verifies Alice's signature using Alice's public key Y_A in the same way as Bob did in the previous step. We can formalize STS steps in the following way:
 - (a) Alice $\rightarrow$ Bob : g^x
 - (b) Alice $\leftarrow$ Bob : $g^y, E_k(S_B(g^y, g^x))$
 - (c) Alice $\rightarrow$ Bob : $E_k(S_A(g^x, g^y))$

We should notice that STS protocol provides forward secrecy as one of the desirable properties in authenticated key exchange protocols. This is achieved due to the Diffie-Hellman protocol which will employ short-term keys for subsequent sessions. The private keys of Alice and Bob are long-term keys and if either one is compromised, the previous sessions are unaffected. Another

property of STS protocol is that it is efficient and does not need redundant exchange of elements with each session due to the use of certificates that shorten the exchange process without consulting a central authority.

Designing secure AKE protocol is a non-trivial problem. Security loopholes in a variety of protocols have been discovered later, after they were adopted. A new type of attack called key compromise impersonation attack (KCI) has been considered to AKE protocol in 1996 by Just and Vaudenay [8]. KCI, as its name indicates, is the situation in which the adversary discloses the private key of either party and be able to masquerade as either legitimate party. Recent AKE protocols are examined to KCI attack. For example, the efficient one-pass asymmetric AKE has been examined by K. Chalkias et al. in 2008 [1].

The Canetti-Krawczyk (CK) model is a formal method to design and analyze secure key agreement protocols by satisfying some desirable security attributes [9]. This model has been extended by LaMacchia et al. [10] to formally consider a new problem called the ephemeral key compromise (EKC) attack. Ephemeral key is a cryptographic key generated once for each key establishment session. EKC involves disclosing the private, ephemeral keys of either party or then computing the session key consequently by the adversary.

There are several key exchange protocols proposed either to improve the secrecy of existing ones or to make new ones that cover a specific need [11] [12] [13] [14] [15] [16]. Complexity, high computational and communication cost of proposed protocols make them not proper for applications that require one way communication. Examples of these applications include email, SMS and printers. In these applications, the receiving party cannot reply instantly or will not reply as in case of printers. To meet the requirements of these new applications, key exchange protocols have been developed [15] [17]. In these protocols, one and only one of the party members are responsible of session key creation. For this reason, these protocols are called one-pass two-party key exchange protocol.

One-pass key exchange protocols are mostly authenticated implicitly. This means the two communicating parties assume that they are the only ones who know the value of the session key and no one else can learn this value. This is considered an advantage for one-pass protocols. However, this type of key exchange protocols cannot provide known key security against the attacker when the later has already recorded a previous run of the protocol and used it for replay attacks. Other drawbacks of one-pass protocols include not supporting perfect forward secrecy (PFS), lack of key control and pruning to key compromise impersonation attacks. The last drawback leads to serious consequences such as reading past/future messages and extracting new information that is

not even exchanged between the parties. This makes studying the attack deeply a motivation to propose a protocol that resists it.

5. One-Pass Protocols

One-pass protocols allow both parties Alice and Bob to establish a session key after sending a single message from Alice to Bob in case an authenticated copy of Bob's public key is already with Alice. Two-pass protocol is made one-pass by replacing Bob's ephemeral public key with his static public key [18]. K. Chalkias et al. [1] have used this technique to convert two-pass protocols to one-pass such as converting Unified Model (UM) [19], Key Exchange Algorithm (KEA) [20] and modified version of KEA(KEA+) [21]. It's assumed that each party among Alice and Bob has already the static private and public keys. The public key is already delivered between party members and verified by each one of them. Based on this assumption, the converted protocols work as follows:

1. Conversion of Unified Model to One-Pass Protocol

- Alice generates an ephemeral key pair (r, R) and sends the public ephemeral key R along with her identity ID_A to Bob. Note that the ephemeral key pair is randomly created for each session. This key is used once and then destroyed as the session ends so that it is not going to be recoverable.
- Bob receives the public ephemeral key from Alice. He computes the session key $K_b = X_B Y_A \oplus X_B \cdot R$
- Alice computes the session key $K_a = X_A Y_B \oplus Y_B \cdot r$
- Conversion of Key Exchange Algorithm (KEA) to One-Pass Protocol
 - Alice generates an ephemeral key pair (r, R) and sends the public ephemeral key R along with her identity ID_A to Bob.
 - Bob receives the public ephemeral key from Alice. He computes the session key

$$K_b = X_B Y_A \oplus X_B \cdot R$$

- Alice computes the session key $K_a = X_A Y_B \oplus Y_B \cdot r$

1. Conversion of Key Exchange Algorithm (KEA+) to One-Pass Protocol

- Alice generates an ephemeral key pair (r, R) and sends the public ephemeral key R along with her identity ID_A to Bob.
- Bob receives the public ephemeral key from Alice. He computes the session key

$$K_b = H(X_B.Y_A, X_B.R \oplus ID_A \oplus ID_B)$$

- Alice computes the session key

$$K_a = H(X_A.Y_B, rY_B \oplus ID_A \oplus ID_B)$$

Now these converted protocols are considered one-pass protocols since they need only one pass to complete key establishment.

4.3 Compromise Impersonation Attack (KCI)

The focus now is to describe and examine the key compromise impersonation attack (KCI) and how it can lead to serious consequences [26].

KCI attacks are too risky as it allows the attacker to get information actively by masquerading as trusted party for the corrupted party. This results in different attack scenarios as following:

- KCI attack lets the corrupted party to receive information and data which threatens him such as viruses, malware and spyware. Usually this content is received by the victim as it is from trusted sources while it is not.
- KCI can cheat the victim by showing the attacker as trusted online store asking for credit information. KCI lets the attacker to impersonate trusted wireless connection which tricks the victim.

Before going into details of this vulnerability on one-pass key exchange protocols, it is important to denote that all public keys are stored in a trusted directory called Certificate Authority (CA). This means all parties can be authenticated.

Assuming one of the parties' private key has been compromised by an attacker. In addition to the ability of the attacker to impersonate the attacked party to others, he can also attack the same party without his knowledge by masquerading as any other party! For this reason, a resistance to key-compromise impersonation attack is a must to reduce the damage that may occur if a private key is disclosed by anyway. Several enhancements are proposed to offer resistance against KCI attacks as in [10] [17] [22][25]

1. Types of KCI Attack

K. Chalkias et al. [1] have classified KCI attacks into two types. In the first type, the attacker masquerades as a trusted party to the victim. This type of attack works as following:

- The attacker, Darth, discloses the private key of Alice already in some way.
- Since Darth knows the private key of Alice, he starts a new session with her showing himself as a trusted party as (Bob) for her. He creates and sends ephemeral public (R) to her.
- Now, Darth acts as if he is Bob to Alice. What makes Alice trust him is that Darth can compute the session key with her because he already knows her private key.

What makes this type of attack possible with one-pass protocols is missing of sender verification, process. If sender is verified by the receiver, he is not going to be able to masquerade as another trusted entity.

There is a possible solution suggested by K. Chalkias et al. to this type of attack. The solution is to force the sender to send his digital signature with his ephemeral public key (R). Moreover, the sender must attach the receiver's identity in the signed message so that the attempt to exploit a protocol run between trusted party and someone else can be avoided. The suggested solution doesn't cover replay attacks. However, it can be reduced by embedding time stamp T so that the receiver examines T. If there is much time passed since the time of protocol run, then he drops the session. Otherwise, he accepts it[25][26].

The second type of KCI attack succeeds with almost all one-pass protocols. In this type:

- The attacker, Darth, has Bob's long term private key in some way.
- If Alice imitates a session with Bob, and if Darth sniffs a message going from Alice to Bob, then Darth can compute the session key between the two parties.
- Darth now is able to impersonate Alice and only Alice to Bob in the current active session only.

Compared to the first type of KCI attack, this type is considered limited as it cannot impersonate any entity at any time. However, there is no obvious solution to this type.

Conclusion

In this paper, a survey on key exchange protocols and possible attacks on the recent ones are presented. We surveyed on secure and efficient key exchange protocols. We presented a key compromise impersonating attack and an ephemeral key compromise attack on recent protocols such as one-pass to test the security resistance factors of each protocol.

References

- [1] K. Chalkias, F. Baldimtsi, D. Hristu-Varsakelis and G. Stephanides, Two Types of Key-Compromise Impersonation Attacks against One-Pass Key Establishment Protocols, in 4th International Conference, ICETE 2007, Barcelona, Spain, July 28-31, 2007, Revised Selected Papers, 2009.
- [2] W. Diffie and M. Hellman, New directions in cryptography, *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644 - 654, 1976.
- [3] W. Stallings, Diffie-Hellman Key Exchange, in *Cryptography and Network Security Principles and Practice*, Pearson Education, 2013, pp. 287-291.
- [4] C. G. Günther, An Identity-Based Key-Exchange Protocol, in *Advances in Cryptology — EUROCRYPT 89*, Springer Berlin Heidelberg, 1990, pp. 29-37.
- [5] E. Okamoto and K. Tanaka, Key distribution system based on identification information, *Selected Areas in Communications, IEEE Journal on* , vol. 7, no. 4, pp. 481-485 , 1989.
- [6] A. Fiat and A. Shamir, How to prove yourself: practical solutions to identification and signature problems, in *Advances in cryptology—CRYPTO 86*, Springer-Verlag London, 1987, pp. 186-194.
- [7] W. Diffie, P. C. V. Oorschot and M. J. Wiener, Authentication and authenticated key exchanges, *Designs, Codes and Cryptography* , vol. 2, no. 2, pp. 107 - 125 , June 1992 .
- [8] M. Just and S. Vaudenay, Authenticated Multi-Party Key Agreement, in *ASIACRYPT '96 Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, 1996.
- [9] R. Canetti and H. Krawczyk, Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels, in *International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings*, 2001.
- [10] B. LaMacchia, K. Lauter and A. Mityagin, Stronger security of authenticated key exchange, in *First International Conference, ProvSec 2007, Wollongong, Australia, November 1-2, 2007. Proceedings*, 2007.
- [11] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kuttan, R. Molva and M. Yung, Systematic Design of Two-Party Authentication Protocols, in *Advances in Cryptology — CRYPTO '91*, 1992.
- [12] S. Blake-Wilson and A. Menezes, Authenticated Diffie-Hellman Key Agreement Protocols, in *SAC '98 Proceedings of the Selected Areas in Cryptography*, Tavares, 1998.

- [13] C. Boyd, W. Mao and K. G. Paterson, Key Agreement Using Statically Keyed Authenticators, in *Applied Cryptography and Network Security*, Jakobsson, 2004.
- [14] O.-R. P. f. T.-P. A. K. Exchange, One-Round Protocols for Two-Party Authenticated Key Exchange, in *Applied Cryptography and Network Security*, Jakobsson, 2004.
- [15] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, An efficient protocol for authenticated Key Agreement, *Designs, Codes and Cryptography*, vol. 28, no. 2, pp. 119 - 134, 2003.
- [16] R. Lu, Z. Cao, R. Su and J. Shao, Pairing-Based Two-Party Authenticated Key Agreement Protocol, 2005,
available: <http://eprint.iacr.org/2005/354>.
- [17] H. Krawczyk, HMQV: a high-performance secure diffie-hellman protocol, in *Advances in Cryptology – CRYPTO 2005: 25th Annual International Cryptology Conference*, Santa Barbara, California, 2005.
- [18] S. Blake-Wilson, D. Johnson and A. Menezes, Key agreement protocols and their security analysis, in *Proceedings of the 6th IMA International Conference on Cryptography and Coding*, Darnell, 1997.
- [19] R. J. D. M. M. Ankney, The Unified Model, in *Contribution to X9F1*, 1995.
- [20] NIST, 1998.
available: <http://csrc.nist.gov/groups/ST/toolkit/documents/skipjack/skipjack.pdf>.
[Accessed 2014].
- [21] T. Kwon, Authentication and Key Agreement via Memorable Password, contribution to the IEEE P1363 study group for future PKC standards, 29 July 2000,
available: <http://eprint.iacr.org/2000/026>. [Accessed 3 11 2014].
- [22] R. W. Zhu, X. Tian and D. S. Wong, Enhancing ck-model for key compromise impersonation Resilience and Identity-based Key Exchange, *Cryptology ePrint Archive: Report 2005/455*, 13 December 2005. [Online]. Available: <http://eprint.iacr.org/2005/455>. [Accessed 3 11 2014].
- [23] H. Elkamchouchi and M. Eldefrawy, An efficient and confirmed protocol for authenticated key agreement, in *Radio Science Conference, 2008. NRSC 2008. National, Tanta, 2008*.
- [24] Q. Cheng, G. Han and C. Ma, Analysis of Two Authenticated Key Exchange Protocols, in *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on, Hubei, 2009*.
- [25] J. Kar “Low Cost Scalar Multiplication Algorithms for Constrained Devices”, *International Journal of Pure and Applied Mathematics*, Vol.102, No.3, pp.579-592, 2015.
- [26] M. R Mishra, J. Kar & B. Majhi, “Practical deployment of One-pass key establishment Protocol on Wireless Sensor Networks”, *International Journal of Pure and Applied Mathematics*, Vol(100), No-4, pp 531-542, 2015