International Journal of e-Collaboration
Volume 15 • Issue 1 • January-March 2019

Towards a Secure Development Environment for Collaborative Applications

Shyam P. Joy, MVJ College of Engineering, Bengaluru, India
Priya Chandran, National Institute of Technology, Calicut, India

ABSTRACT

Collaborative applications use the security services offered by secure socket layer / transport layer security (SSL/TLS) to implement authentication and confidentiality. Since SSL/TLS establishes a secure communication between two participants, for a secure network of n (> 2) participants, at least n(n-1)/2 secure communication channels have to be established. Whereas, a group key agreement (GKA) protocol allows the participants to compute a common secret group key as a function of the secrets of participants, and thereby remove the n(n-1)/2 lower bound on the channel requirement. Partial forward secrecy is a property of the GKA protocol which assesses the secrecy of the group key, when the secrets are compromised. Collaborative applications have different security requirements. Hence, the Spread Toolkit offers a set of GKA protocols, so that the designers can choose the most appropriate one. In this article, given a set of GKA protocols, a method is proposed to select the best among them, with respect to partial forward secrecy.

KEYWORDS

GKA Protocols, Group Key Agreement Protocols, Partial Forward Secrecy, Spread Toolkit

DOI: 10.4018/IJeC.2019010101

INTRODUCTION

Collaborative software applications enable members of a group to communicate with each other and achieve common objectives. Enterprises use social networks as collaborative networks to brainstorm participants to improve the knowledge level within the organization (Franchi et al., 2013). Computer supported collaborative learning provides support for coordinating co-operation between teachers and students, in learning activities. Collaborative Security intends to secure the Internet by coordinating security activities of users (Meng et al., 2015). The communication between the participants in the applications mentioned above need not be at the same time; or in other words the communication can be asynchronous. On the other hand, multimedia conferencing and multi-user games are synchronous collaborative applications. Vehicles in a network could also be participants in a conference.

Conferencing applications are collaborative applications which allow geographically separate users to exchange multimedia information in real-time. Conferencing applications should provide security properties such as authentication and confidentiality. Currently, conferencing applications like Adobe Connect, BigMarker, Cisco WebEx, and Skype use SSL/TLS for providing security. SSL/TLS is a security protocol to establish a secure communication channel between two participants. SSL/TLS provides authentication by means of public key certificates, using Public-Key Infrastructure, and confidentiality by symmetric key encryption using Advanced Encryption Standard (AES). The cryptographic algorithms for symmetric encryption and authentication are negotiated during handshaking sub-protocol of SSL/TLS.

The use of SSL/TLS for setting up secure collaborative networks would need additional communication overheads because at least n($n-1)/2$ handshakes are needed, for securely interconnecting n participants using SSL/TLS. Since, each handshake protocol of SSL requires four messages, at least $4n(n-1)/2$ messages are needed.

GKA protocols (Manulis, 2008) enable each member of a group to compute a common secret, referred to as group key. The number of messages needed for GKA protocol to establish a secure network is less compared to SSL/TLS. For example, to interconnect n participants, Group Diffie-Hellman (GDH) protocol (Ateniese et al., 2000), Burmester and Desmedt (BD) protocol (Burmester & Desmedt, 1994) and a protocol by D. G. Steer, L. Strawczynski, W. Diffie, and M. Wiener (SLDW) (Steer et al., 1990; Kim et al., 2002) takes $2(n-1)$ messages, while Tree Based Group Diffie Hellman (TGDH) protocol (Kim et al., 2004b) takes $2n$ messages. Moreover, GKA protocols are more suited for peer-to-peer architecture than SSL/TLS (Amir et al., 2002; Liu & Koenig, 2005).

GKA protocols compute the group key as a function of contributions of participants. The contributions are secrets of the participants. The hash of the group key is further used for symmetric encryption of the messages exchanged among the group members. The contributions of each group member used in computing the group key in a GKA protocol may have a life term as much as the session, while others may be valid across sessions. Ideally, a GKA protocol must ensure that, the computed group key remains