Poster: A Distributed Security Architecture for P2PSIP

Rasib Khan, Ragib Hasan
SECRETLab@UAB, Department of Computer and Information Sciences
University of Alabama at Birmingham, AL, USA
Email: {rasib, ragib}@cis.uab.edu

***Abstract*—Since early days, peer-to-peer (P2P) protocols have proven significant improvements over traditional client-server models. On the other hand, the Session Initiation Protocol (SIP) is a popular protocol for establishing multimedia sessions, and uses a client-server approach. P2PSIP is an architecture for deploying SIP services over a P2P network overlay, and thus leverages the limitations of client-server architectures for SIP. In this work, we present a novel scheme for a secured and distributed P2PSIP model. Our architecture allows a ad-hoc deployment, with a scalable design for a completely distributed security infrastructure.**

***Keywords*-Chord; Distributed Security Model; P2PSIP; Shamir's Secret Sharing;**

I. Introduction

Peer-to-peer (P2P) architecture is popular since the late 90's. P2P networks introduces significant advantages over traditional client-server models. The basic feature of peers being requesters as well as providers is the most attractive feature of P2P networks. The Session Initiation Protocol (SIP) is a signaling protocol for establishing media sessions. SIP is widely used for VoIP services and requires a client server architecture.

However, the centralized nature of SIP is a significant bottleneck for deployment and performance. This requirement has given birth to the concept of SIP services over P2P network architectures, also known as P2PSIP. P2PSIP replaces the client-server model by removing the centralized SIP server. Thus, P2PSIP enables fast set-up, ad-hoc formation, easy deployment, and robustness against failures [1], [2].

Nonetheless, security in P2PSIP suffers from many shortcomings in its current implementations. So far, multiple solutions have been put forward to mitigate the security issues in P2PSIP [3], [4]. The decentralized architecture in P2PSIP reduces the management and hence introduces multiple security issues. The existence of malicious peers in the P2P overlay creates a significant issue in ensuring a secured environment. Thus, a malicious peer will be able perform man-in-the-middle attacks by dropping, tampering, and re-routing SIP messages between two other peers. Certain architectures [5], [6] apply the public key certificates to ensure mutually authenticated peers for SIP sessions, and guarantee confidentiality and authenticity of SIP packets. Another solution for securing P2PSIP is by using cryptographically generated SIP URIs to authenticate nodes when starting a session [4]. Seedorf *et. al*. in [7] and [8] demonstrates the possible attacks on P2P overlays for SIP.

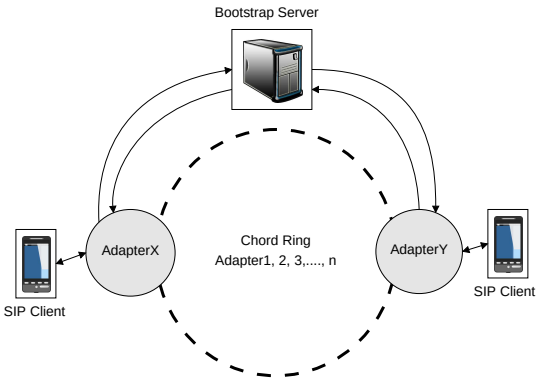

Fig. 1: Architecture of the secured P2PSIP overlay

However, most research works on P2PSIP have focused on distributing address of records for name resolution. Security researches enforced public key infrastructures and security through obfuscation. As a result, they introduced a bottleneck in the architecture or involved external network connectivity, thus removing the feature of ad-hoc deployments for P2PSIP systems.

In this work, we present the design for a distributed security architecture for P2PSIP. The proposed scheme allows ad-hoc deployment for the SIP based services, with a distributed model for the secured SIP session establishment. Decentralization of the security mechanism allows the scheme to be scalable and without any single point of failure. We claim the decentralized mode of security features makes the system resilient against malicious peers on the P2P overlay network. Thus, the process of name resolution and SIP session agreement is secure and tolerant of the randomness of P2P systems.

II. A Distributed Secured P2PSIP Architecture

In this section, we present a simple and scalable architecture for distributed security management for SIP over a p2p Chord overlay network.

A. System Model

The proposed architecture includes the following entities. The **Chord Overlay** is a ring of independent and distributed nodes. The Chord overlay network is initially created by the **Bootstrap Server**. The bootstrap server is the only central entity in this architecture, and the nodes contact the bootstrap server to obtain the details for joining the overlay. The **Secured P2PSIP Adapter** is a service running on a user device. The

service acts as a bridge between the user and the Chord overlay network to provide secured services for SIP call establishment. Finally, there is the off-the-shelf **SIP client** which implements the RFC-3265 standards. We illustrate the overview of the system in figure 1.

B. Secured P2PSIP Registration

The registration process for the secured P2PSIP architecture includes two phases.

The SIP client tries to register with the local adapter. The adapter sends the public key along with the registration request to the bootstrap server. The bootstrap server receives the register request, and executes the Shamir's Key Sharing Algorithm [9] to divide the public key for the user into n pieces (*PubPeyPiece*), with at least k pieces required for reconstructing the public key. Here, the values of k and n are flexible, and can be chosen according to the preferred security level.

The bootstrap server then stores the n pieces of the public key in the Chord overlay, on the nodes currently present within the Chord ring. Next, the bootstrap server stores the adapter lookup information in the Chord overlay network.

Finally, the bootstrap server responds to the registering adapter with a SECP2PSIP REGISTERED. Additionally, the bootstrap server also includes the Chord overlay information with the registration response. Upon receiving the success response from the bootstrap server, the adapter then joins the Chord overlay network, and sends a standard *200 OK* response to the SIP client.

C. Secured P2PSIP Call Establishment

The process of establishing a SIP session in the P2PSIP architecture requires two individual phases.

At the beginning, *userA* makes a standard SIP INVITE request to *sip:userB@userA.ip.address:adapterA.port*. The local adapter receives the INVITE and performs a Chord table lookup to find *userB* from the overlay network. The adapter randomly retrieves k pieces of *userB*'s public key from the Chord overlay. The public key for *userB* is reconstructed using Shamir's Key Sharing Algorithm [9]. Subsequently, *adapterA* then creates a SECP2PSIP INVITE message, and sends it to *adapterB*.

AdapterB receives the SECP2PSIP INVITE message, and validates the request. At first, *adapterB* constructs *userA*'s public key in a similar manner as mentioned earlier for *adapterA*. Once *adapterB* verifies the authenticity of the SECP2PSIP INVITE, *adapterB* responds to *adapterA* with a SECP2PSIP REDIRECT.

AdapterA receives the SECP2PSIP REDIRECT request. Once the authenticity and integrity of the SECP2PSIP REDIRECT is successfully verified, *adapterA* sends a SIP/2.0 302 MOVED TEMPORARILY request to *userA*'s SIP client. The SIP client receives the message, and redirects the SIP call to *userB*'s SIP client. The redirection of the SIP client and the subsequent behaviors occur according to standard SIP definition [10], without the interference of the Chord overlay in the process.

III. Conclusion

P2P systems are better in comparison to centralized architectures in terms of scalability. P2PSIP utilizes the advantages of an overlay network to provide SIP based services. However, the presence of malicious peers in the network and lack of management makes the systems unsecured and thus vulnerable to attacks. Although substantial research has been done in securing P2PSIP, the distributed feature for P2PSIP has not been the primary concern while implementing the secured architectures [4], [3], [11], [12]. The proposed scheme in this work utilizes the Chord overlay P2P network and Shamir's Key Sharing Algorithm [9] to diffuse the risk of public keys stored on the overlay. Hence, an attacker is required to control at least k out of n peers to interfere in the process of establishing a secured P2PSIP session.

References

[1] D.A. Bryan, E. Shim and B.B. Lowekamp, "Use Cases for Peer-to-Peer Session Initiation Protocol (P2P SIP)," *Internet draft, http://www.p2psip.org/drafts/draft-bryan-sipping-p2p-usecases-00.html*, Nov 2005.

[2] P2PSIP IETF Working Group, "http://www.ietf.org/html.charters/p2psip-charter.html, last accessed 15th may 2001."

[3] Xianghan Zheng and Vladimir A. Oleshchuk, "The Design of Secure and Efficient P2PSIP Communication Systems," *Proceedings of WISTP*, pp. 253–260, 2010.

[4] I. Baumgart, "P2PNS: A Secure Distributed Name Service for P2PSIP," *Sixth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2008*, pp. 480–485, Mar 2008.

[5] C. Jennings, B. Lowekamp, E. Rescorla, S. Baset, and H. Schulzrinne, "Ietf draft, REsource LOcation And Discovery (RELOAD) Base Protocol," Mar 2010.

[6] J. Hautakorpi, and G. Schultz, "A Feasibility Study of an Arbitrary Search in Structured Peer-to-Peer Networks," *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1 – 8, Sept 2010.

[7] J. Seedorf, F. Ruwolt, M. Stiemerling, and S. Niccolini, "Evaluating P2PSIP under Attack: An Emulative Study," *Global Telecommunications Conference,IEEE GLOBECOM*, pp. 1–6, Nov 2008.

[8] J. Seedorf, "Security challenges for peer-to-peer SIP," *Network, IEEE*, vol. 20, Number 5, pp. 38–45, Sept 2006.

[9] A. Shamir, "How to share a secret," *Commun. ACM, http://doi.acm-org/10.1145/359168.359176*, vol. 2, Issue 11, pp. 612–613, Nov 1979.

[10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP - Session Initiation Protocol," 2002.

[11] X. Zheng and V. Oleshchuk, "A secure architecture for p2psip-based communication systems," pp. 75–82, 2009. [Online]. Available: http://doi.acm.org/10.1145/1626195.1626216

[12] X. Zheng and V. Oleshchuk, "Trust enhancement of p2psip communication systems," *International Journal of Internet Technology and Secured Transactions (IJITST)*, vol. 3, no. 2, pp. 121–133, April 2011. [Online]. Available: http://dx.doi.org/10.1504/IJITST.2011.039773